



user's guide

hp StorageWorks data exchange



Notice

© 2002, Hewlett-Packard Company.

Part Number B7914-96002

E0702

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard. The information contained in this document is subject to change without notice.

Warranty

If you have any questions about the warranty for this product, contact your dealer or local HP sales representative.

Trademarks

Windows NT, Microsoft, Windows, MS Windows, and MS-DOS are U.S. registered trademarks of Microsoft Corporation.

Other reserved names are trademarks of the respective companies.

Safety notices

Any servicing, adjustment, maintenance, or repair must be performed only by authorized service-trained personnel.

Caution Denotes a hazard that can cause hardware or software damage.

WARNING Denotes a hazard that can cause personal injury or death.

Format conventions

literal Specifies text to be typed exactly as shown, such as commands, path names, file names, and directory names.

variable Indicates that you must supply a value.

output Denotes text displayed on the screen.

Revision History

June 1997	Initial edition
August 1997	Correction and addition
	Correction of Index VTOC supporting
	Correction of volume name to “3390-3B”
	Correction of OS version and motif version
	Correction of volume name to “3390-3B”
	Correction and addition of installation procedure
	Correction of parameter and volume definition files
	Correction of error code and message
	Addition of operational notes
	Addition of Troubleshooting section
September 1997	Addition and correction relating to Data Exchange OTM, emulation types expansion and Windows NT supporting
	Addition of installation procedure for Windows NT
	Addition of restriction on parameter definition file
	Addition of warning

	Correction of window display
	Addition of operational note
	Correction and addition of error code
March 1998	Correction of each description
March 1998	Consolidation of FAL and FCU manuals
	Addition of description for RAID300
	Addition of description for IO contention
	Addition of description for Digital UNIX
	Addition and correction of each description
September 1998	Update of supporting program version
	Adding supported OS versions
	Addition of notes for thread safe and reserved words
	Addition of note for error in NT
	Correction of function format
	Correction of error code for FAL
	Change for supporting DLL
	Correction of record
	Addition of record length adding function
	Addition of note for bidirectional transfer
	Addition of label writing for Data Exchange volume
	Change for new installation procedure

September 1998	Addition for NCR support
	Addition of empty file and RDW parameters
	Change for direct parameter setting
	Delete unnecessary description
	Addition of note for overwrite operation
	Change for Windows NT GUI improvement
	Addition of version display
	Change of FCU error code for UNIX
	Addition of FCU error code for Windows NT
	Correction and addition of operational notes
November 1998	Addition of troubleshooting for Windows NT
	Change revision of this document and FAL version
	Addition of note for Digital UNIX device name
December 1998	Addition of note for a comment line
	Update of supported functions
	VSE support
	Code conversion table file support
	Windows NT screen changes for supporting VSE and code conversion file
	DYNIX/ptx support
	Comment line support for UNIX

	Error code changes
	Editorial changes
	Addition of Fibre Channel
	Correction of supporting DYNIX version
	Addition of notes on error code
	Correction code conversion table
	Addition of de-installation procedure for new version code
	Other corrections
February 1999	Addition of Data Exchange OTO support
	Addition of C-type volume support
	Addition of Data Exchange OTM data padding support
	Addition of notes for FAL/FCU operations
	Addition of LISTVOL accessory function
March 1999	Correction of code conversion for Data Exchange OTO
	Addition of remarks for FC connection
	Correction of host combination list for Data Exchange OTO
	Correction of description for label writing option
	Correction of EMP to Emp
	Correction of dataset name for “?”.
April 1999	Delete NCR4 3.01 from Table 6

June 1999

Addition of pipe function support

Change the contents of Table 7

Add 5.2 pipe function

Add explanation

Add explanation in (4)

Add the field of PIPE (9)

Add explanation in (10) and note

Add note

Change copyright (1998 – 1999)

Add explanation in volume

Add a toggle button

Add section 6.5.1.8

Add explanation in code conversion

Change 5.4 – 5.5

Add notes in VSE and status

Add notes in VSE

Change the OTO (Format/Allocate) function

Addition of error code (-132/-250-- -254/ -390-- -398)

Addition of error code (-111-- -115/-121-- -124/ -245--
-307)

Add the sentence (16)--(18)

June 1999	<p>Add explanation</p> <p>Add Formatter and Allocator error message</p> <p>Change 5.3à5.4</p> <p>Add contents</p> <p>Change block length 5-->9</p>
August 1999	<p>Add “PIPE=”</p> <p>Add explanation</p> <p>Change Note ->Next</p> <p>Change Fig.6.5.2-81</p> <p>Change Exit (Ctrl+X) ->Exit</p> <p>Change explanation (16)-(18)</p> <p>Add (20)</p>
January 2000	<p>Add version 01-01-44/00</p> <p>Delete note 14</p> <p>Add OPEN-8/9/K</p> <p>Add datasetPut2</p> <p>Add winerror.h</p> <p>Add -2 and -6</p> <p>Change S ->s, T -> t</p> <p>Change Fig.6.5-2</p> <p>Add “Ek<->J”, Change EA -> Ec/E<->A</p>

	Change main dialog box and option dialog box
	Add VB and FB
	Delete (13) and (18) . Add (18), (19) and (20)
	Add OPEN-8/9/K
	Add -26
	Add EcA
	Delete note in status
	Add explanation in VSN
	Delete EA in -107
	Delete explanation in -230 and -105
	Change (15)
	3331 – 5818 OPEN-3 -> OPEN-3/8/9/K
February 2000	Add (21), (22)
April 2000	Add server machine for RAID400
	Add #46
	Add * in SCSI
	Add -35/-36/-37
	Add 6.1.3
	Delete (9) note, add (23)
May 2000	(1) Add remarks
	Delete (*)

	Add rm ppkeyset
	Add VSE remark
	Add VSE remark
July 2000	Add Digital UNIX
	Add explanation
	Add “or upper”
	Add 4369 cylinders
July 2000	Add 01-02-47/00 and 01
	Fibre => Fiber
	Add Solaris 7
	Delete (4)
	Delete note
September 2000	Add #48
	Add OPEN-E/M
	Add Solaris 8
	Add note
	Change data cylinder extent
	Add VSN identification
	Add DYNIX/ptx
	Add (23) and (17)
	Add -18

	Add -124,-399
	-125,-126
	Change number of cylinders
	Delete spin button
	Add VSN identification
	Add (24)
	Add (2) [windows] iii)
December 2000	Move Step 3
	Change bytes
	Table 10.3.1 -> 10.3.2
	1024->1025, 6->7, 10->11
	Add Tru64 UNIX
	Add Tru64 ver5.0
February 2001	Add (13)
March 2001	Add #50
	Change note 2
	Add maximum data size
	Add 5.7
	Add 6.2.2
	Add 6.2.1
	Add total of volume

Change -18

Change -26,-27

Add -39- -48

Change -121/-300

Add (25), change (13)/(17)

Add install method (Tru64)

June 2001

Add #51

Add OPEN-L & add/change explanation

Add (26)

Add PIPE (NT: not available)

Add error message (cylinders large error)

Updates

May 2002

update content to v.15; convert content to HP template;
create separate chapters for FCU operations for UNIX
and Windows NT

For the most current information about HP StorageWorks products, visit the
support web site:

www.hp.com/support/stressfree

For information about product availability, configuration, and connectivity,
consult your HP account representative.

CONTENTS

Revision History 3

1 Overview 7

2 System Configuration 13

Administration Notes 13

Functional Position of File Access Library and File Conversion Utility 14

Supported Objects 16

Maximum Data Capacity 16

Host (Mainframe) OS 17

Host (Mainframe) Datasets 17

Mainframe Device Emulation Types 19

Data Transfer Directions 21

Pipe Functions 22

Code Systems Supported 22

Server Machines Supported 22

3	File Access Library (FAL) Functions	25
	Opening a Dataset	26
	Reading Data	28
	Writing Data	30
	Closing a Dataset	32
	Acquiring Error Information	32
	Acquiring Dataset Attribute Information	33
	Error Information	39
4	How to Use File Access Library (FAL)	47
	Providing Operational Environment	48
	Shared Volume Definition	48
	Creating a Volume Definition File	48
	Installing File Access Library	48
	Compiling File Access Library	48
	Using the File Access Library Application Program	51
	Example: Reading Data	51
	Example: Acquiring Dataset Attributes	52
5	Functions	53
	File Conversion Functions	54
	Data Editing Formats	54
	Pipe Functions	56
	Time Out Value Definition	56
	Code Converting Functions	58
	Using the Code Conversion Table Included in FCU	58
	Using the User-defined Code Conversion Table	58
	Padding Functions	60
	Variable Length MTO Data Transfer	60
	Fixed Length OTM Data Transfer	61
	Delimiter Adding Functions	62
	Data Editing Formats	62
	Record Length Adding Function	66

Data Exchange from Multiple Volume Datasets	67
Bi-directional Data Transfer between Mainframe and Open Systems	68

6 How to Use File Conversion Utility (FCU) 69

Providing Operational Environment	70
Shared Volume Definition	70
Installing the FAL/FCU Program for UNIX	72
Installing the FAL/FCU Program for Windows NT	79
Inputting License Keys	81
Setting License Keys	82
Definition Files	83
Volume Definition Files	83
Multiple Volume Definition Files	93
Parameter Definition Files	95
Starting the File Conversion Utility	105
Starting FCU for UNIX	105
Starting FCU for Windows NT	107

7 FCU Operations and Screens for Windows NT 109

FCU Interface	110
Splash Screen	110
Main Screen	110
Tool Bar	121
Buttons and Fields	123
Status Bar	124
FCU Operations	125
Creating Volume Definition Files for FCU	125
Starting the FCU	126
Creating New Parameter Definition Files	128
Entering Parameter Line Data	128
Saving Parameter Definition Files to Disk	136
Opening Existing Parameter Definition Files	137
Executing Parameter Lines	137

8	FCU Operations and Screens for UNIX	139
	Copyright Screen	140
	Operation Screen	141
	Operation Screen Menus	142
	Operation Screen Buttons, Check Boxes, and Fields	147
	Non-window FCU Operations	154
	Indication and Confirmation of Detail Parameters	154
	File Verification Display	154
	Confirmation of Overwrite	154
	File Conversion Process Initiation	155
	File Conversion Process Results	155
	Completing the FCU Process	155
	LISTVOL	156
9	FCU Error Information	159
	UNIX FCU Errors	160
	Windows NT FCU Errors	178
10	Operational Notes	187
	FAL/FCU Operations	188
	IO Contention between Mainframe and Open System	193
11	Troubleshooting	195
	Troubleshooting for UNIX	196
	Troubleshooting for Windows NT	197

12	Data Exchange OTO Operation	199
	Providing the Data Exchange OTO Operational Environment	200
	Defining RAW Devices	200
	Installing Formatter (FMT) and Allocator (ALC)	201
	Creating Intermediate Volumes	202
	Creating Intermediate Files	208
	Creating Volume Definition Files	216
	Creating Parameter Definition File	217
	Starting Data Exchange Operation	218
	Formatter/ Allocator Error Information	219
	Errors in UNIX	220
	Errors in Windows NT	223
	Index	225

OVERVIEW

File Access Library and File Conversion Utility (FAL/FCU) is a program which is installed in open system servers and provides functions for exchanging data between a mainframe host and an open system host, or between open system hosts.

FAL/FCU has the following features:

- Application programs (APs) can read and write file data unconsciously of data format (e.g. CKD format which is specific to the mainframe data)
- Data is directly read and written via XP256/512 disk subsystem, making the data transfer faster than that through the network
- Application programs described in C can use functions of the File Access Library (FAL), because the functions are provided as a library (functions) of C
- You can execute data entities code conversion if needed (e.g. from EBCDIC to ASCII) when a data transfer is performed
- (*UNIX only*) You can transfer data entries from the mainframe to the application program or the utility program on open system directly using a named pipe

Table 1 shows the major changes in each FAL/FCU program version.

Table 1. FAL/FCU Program Version Updates

FAL/FCU Program Version	Reference Manual	Supported Function	Open Platform	DKC Main Micro
Ver. 41.01.02 and above	R7.0 Feb. 5, 1999	<ul style="list-style-type: none"> • Data Exchange OTO • Data Exchange OTM data padding • C-type Data Exchange volumes • List Vol. Accessory function 	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX • IBM/AIX • Windows NT • Digital UNIX • NCR SVR4 • SEQUENT DYNIX/ptx 	XP256
Ver. 42.00.00 and above	R8.0 Jun. 20, 1999	<ul style="list-style-type: none"> • Data Exchange MTO PIPE function support <p><i>Ignore code conversion when Data Exchange OTO</i></p> <p><i>The interchangeability of formatter/allocator</i></p>	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX • IBM/AIX • Windows NT • Digital UNIX • NCR SVR4* • SEQUENT DYNIX/ptx* 	XP256
Ver. 44.11.00 (NT) and above	R9.0	<ul style="list-style-type: none"> • OPEN-8/OPEN-9/ support 	<ul style="list-style-type: none"> • HP/HP-UX • IBM/AIX 	XP256
Ver. 44.01.00 (UNIX)		<ul style="list-style-type: none"> • A record with no data entry support • Display 'VB' and 'FB' in LISTVOL, MF-file information and MF-file select OPEN-Ko SUN/Solaris 	<ul style="list-style-type: none"> • Windows NT 	

Table 1. FAL/FCU Program Version Updates (continued)

FAL/FCU Program Version	Reference Manual	Supported Function	Open Platform	DKC Main Micro
Ver. 46.01.05 (NT)	R10.2	<ul style="list-style-type: none"> • Windows 2000 support • SUN/Solaris 	<ul style="list-style-type: none"> • HP/HP-UX • IBM/AIX 	XP256
Ver. 46.01.03 (UNIX) and above			<ul style="list-style-type: none"> • Windows NT/2000 • Digital/Tru64 UNIX • NCR SVR4* • SEQUENT DYNIX/ptx* 	
Ver. 46.02.05 (NT)	R10.2	<ul style="list-style-type: none"> • XP512 support • License key supports 16 characters 	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX 	XP512
Ver. 46.02.03 (UNIX) and above			<ul style="list-style-type: none"> • IBM/AIX • Windows NT/2000 	
XX.02.XX is XP512 only				
Ver. 47.02.01 (NT)	R11.0	<ul style="list-style-type: none"> • License key supports both 16 and 20 characters 	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX 	XP512
Ver. 47.02.00 (UNIX) and above			<ul style="list-style-type: none"> • IBM/AIX • Windows NT/2000 • Digital/Tru64 UNIX • SEQUENT DYNIX/ptx 	

Table 1. FAL/FCU Program Version Updates (continued)

FAL/FCU Program Version	Reference Manual	Supported Function	Open Platform	DKC Main Micro
Ver. 48.XX.21 (NT)	R12.0	<ul style="list-style-type: none"> • Support LUSE/OPEN-E/OPEN-M 	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX 	XP256/XP512
Ver. 48.XX.20 (UNIX) and above		<ul style="list-style-type: none"> • Support large files (files larger than 2GB and volumes larger than 5818 cylinders) • Support same VSN in volume definition file • Support Solaris 8 	<ul style="list-style-type: none"> • IBM/AIX • Windows NT/2000 	
Ver. 49.XX.00	R13.1	<ul style="list-style-type: none"> • Support Tru64 UNIX v5 (SCSI only) • Change product name (FAL for Tru64 UNIX<- 	<ul style="list-style-type: none"> • Digital UNIX • Compaq/Tru64 UNIX 	XP256/XP512
Ver. 50.XX.YY and above	R14.0	<ul style="list-style-type: none"> • Support multiple volume dataset (Data Exchange MTO only) 	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX • IBM/AIX • Windows NT/2000 • Compaq/Tru64 UNIX 	XP256/XP512
Ver. 01-01-51/YY and above	R15.0	Support OPEN-L/M	<ul style="list-style-type: none"> • SUN/Solaris • HP/HP-UX • IBM/AIX • Windows NT/2000 	RAID200: v.03-16-14 and above RAID300: v. 52-34-xx and above* RAID400

*: FAL/FCU for NCR SVR4 and SEQUENT DYNIX/ptx are only supported in RAID300 and only with the latest DKC micro-code, not with the Ver. 52-34-xx. The supporting DKC micro version will be informed by other notices.

SYSTEM CONFIGURATION

Administration Notes

- The Data Exchange volumes accessed by FAL/FCU belong to mainframe devices.
- You can only access Data Exchange datasets through the FAL/FCU program is allowed for open systems.
- IO contention between mainframe and open systems must be managed by administration level users operation. See “FAL/FCU Operations” (page 188) for more detailed operational notes.
- Job coordination is assumed to be taken at the system level, such as one via LAN.

Functional Position of File Access Library and File Conversion Utility

Figure 1 and figure 2 (page 15) show different system configurations using FAL/FCU and their respective positions in each environment.

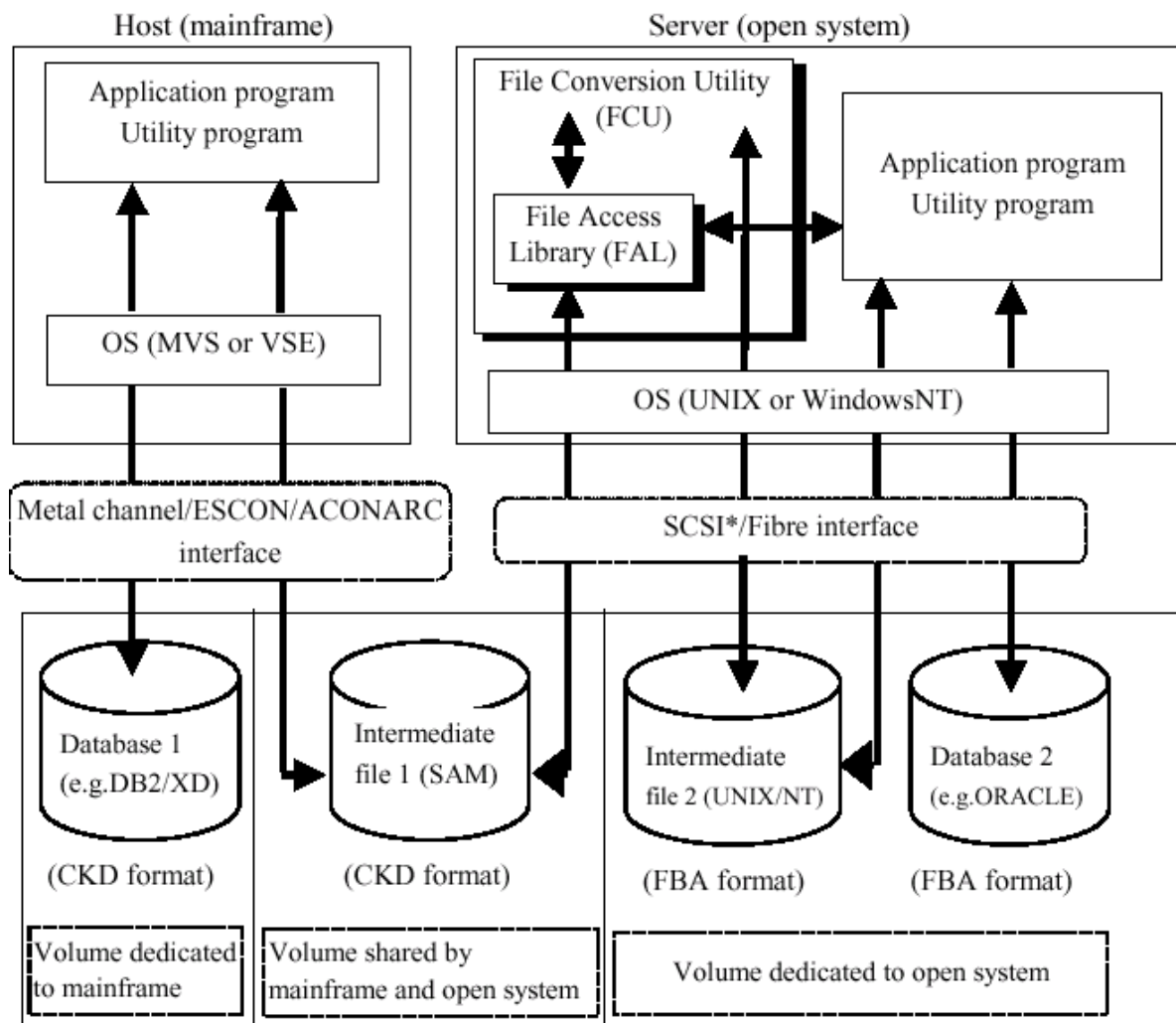


Figure 1. XP256/512-Multiplatform/ Data Exchange MTO and Data Exchange OTM

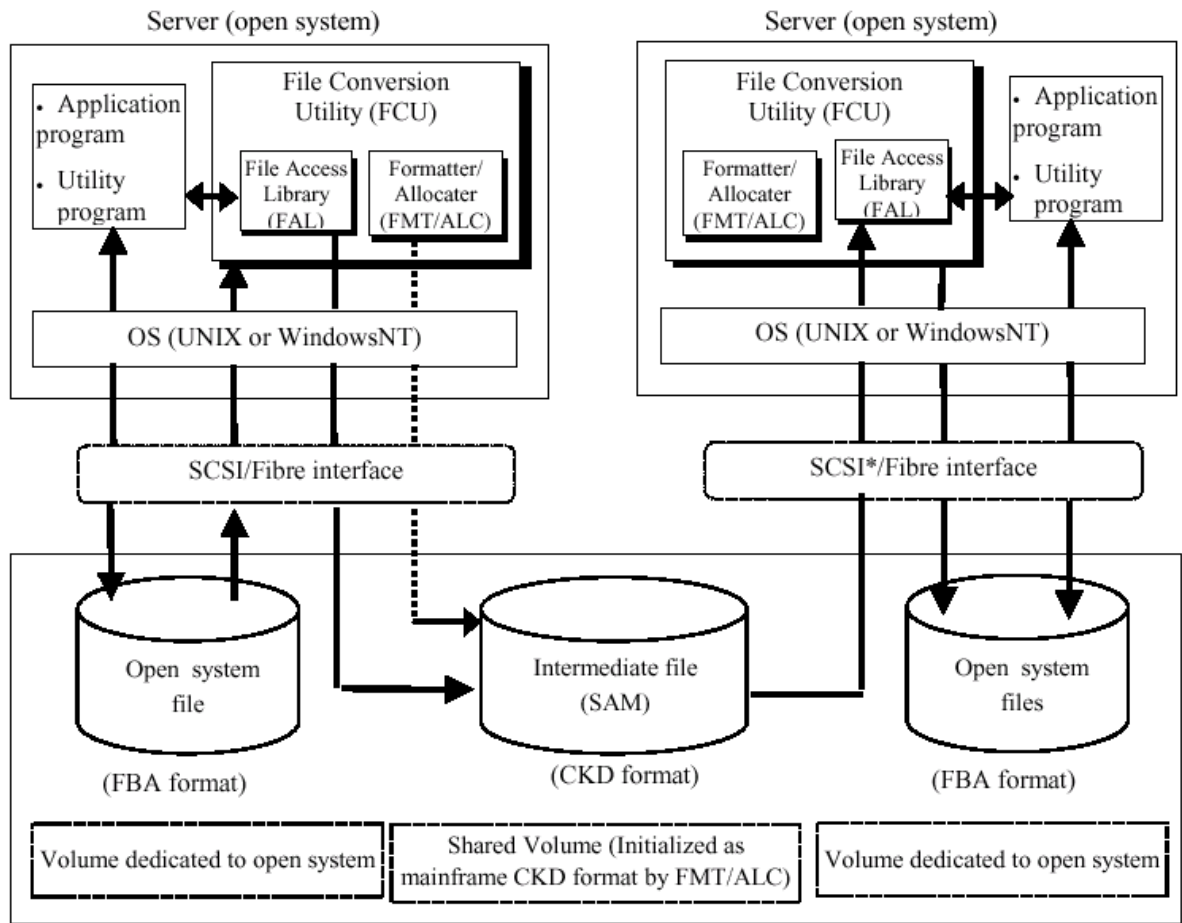


Figure 2. XP256/512-Multiplatform/ Data Exchange OTO

Supported Objects

FAL/FCU supports the following types of objects. This section describes each supported object in more detail:

- Host (Mainframe) OS (page 17)
- Host (Mainframe) Datasets (page 17)
- Mainframe Device Emulation Types (page 19)
- Data Transfer Directions (page 21)
- Pipe Functions (page 22)
- Code Converting Functions (page 22)
- Server Machines (page 22)

Maximum Data Capacity

For versions 48.XX.YY (XX=01 or 02, YY>=20) or above, the intermediate file data capacity is smaller than its physical capacity and varies depending upon the block length to be used.

Table 1. Maximum Data Capacity

Emulation Type	No LUSE	LUSE	For version 50.XX.YY or above XX:01 or 02,YY>=20 using multiple volume datasets
OPEN-3	About 2.1GB	About 42.8 GB	
OPEN-8	About 6.5 GB	About 42.8 GB	
OPEN-9	About 6.5 GB	About 42.8 GB	
OPEN-K	About 1.6 GB	About 42.8 GB	
OPEN-E	About 12.9 GB	About 42.8 GB	
OPEN-L	About 32.3 GB	About 42.8 GB	
OPEN-M (Windows 2000 only)	About 41.9 GB	About 42.8 GB	

Table 1. Maximum Data Capacity (continued)

Emulation Type	No LUSE	LUSE	For version 50.XX.YY or above XX:01 or 02,YY>=20 using multiple volume datasets
3390-3X	About 2.6 GB		HP-UX (*1)/Solaris/NT/Tru64: 78 GB AIX: 64 GB (The limit OS)
3380-KX	About 1.7 GB		HP-UX (*1)/Solaris/Tru64/ NT/AIX: 51 GB

(*1) An open system can only recognize 8 volume/1 fibre board.

Host (Mainframe) OS

This utility supports the following host (mainframe) OSs:

- MVS
- VSE (Not supported in XP512)
- VOS3 (Hitachi OS)

Host (Mainframe) Datasets

The following datasets can be processed in this utility.

Dataset type:	SAM
Record format:	Fixed record or variable record (see Note 13 for VSE)
Block length:	Any length within the extent supported by the OS (see Note 13 for VSE)
Record length:	Any length within the extent supported by the OS (see Note 13 for VSE)

Notes

1. The program supports datasets with standard VTOCs. Datasets with Index VTOC can also be supported, but the index is ignored and the whole VTOC is referred sequentially.
2. *(50.XX.YY and below only)* Multiple volume datasets are not supported. When a dataset is spanned over multiple volumes, only the portion of the dataset within one volume will be processed.
(50.XX.YY and above only) Multiple volume datasets are supported only for Data Exchange MTO.
3. Records with keys are not supported. If a record has a key, an error will be returned.
4. Only the track format managed with standard R0 is supported.
5. Datasets other than SAM (i.e. DAM, PAM, ISAM, VSAM) are not supported and will return an error.
6. Record lengths and spanned record must be specified.
7. When an illegal block length or data with illegal record length is detected, the process will return an error.
8. A dataset over one volume length in the mainframe cannot be processed.
9. Dataset type, record format, block length, and dataset size can be verified by clicking Help | MF-File, or through the LISTVOL command.
10. Dataset name cannot contain spaces.
11. Direct access to a database file is not supported. When you need to process a database file, convert it to SAM file prior to using this function.
12. To use Data Exchange OTM, you must first process the dataset so that it is registered in VTOC; complete allocation of the extent; and specify the attributes, such as dataset type, record format, block length and record length. Because the extent is not expanded automatically by this program, make sure you have enough to store the data from the open system.
13. To use VSE for a mainframe OS, you must first set the record format, record length, and block length in the FCU parameter, because they are not defined in the VTOC. In addition, the VSE datasets cannot be processed by the FAL program if FAL is called from a user program.

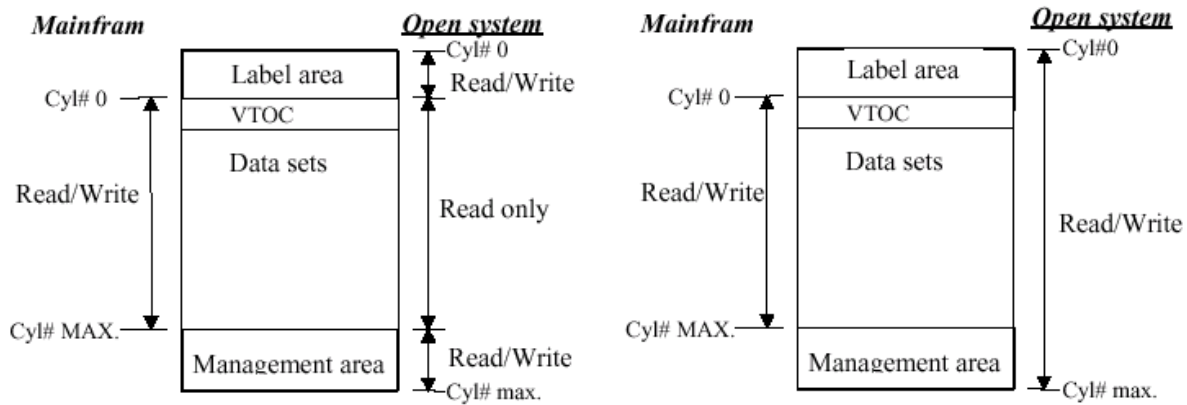
Mainframe Device Emulation Types

FAL/FCU supports the following device emulation types as Data Exchange volumes:

Table 2. Mainframe Device Emulation Types

Device Emulation Type	Use	Mainframe Host Support	Open System Host Support
3390-3A	Data Exchange MTO and Data Exchange OTM	same access as for 3390-3 is allowed	read/write
3390-3B	Data Exchange MTO	same access as for 3390-3 is allowed	read only
3390-3C	Data Exchange OTM	same access as for 3390-3 is allowed but read only	read/write
3380-KA	Data Exchange MTO and Data Exchange OTM	same access as for 3380-K is allowed	read/write
3380-KB	Data Exchange MTO	same access as for 3380-K is allowed	read only
3380-KC	Data Exchange OTM	same access as for 3380-K is allowed but read only	read/write
OPEN-3/ OPEN-8/ OPEN-9/ OPEN-K/ OPEN-E/ OPEN-M/ OPEN-L	Data ExchangeOTO (after initializing with FMT and ALC)	no access	read/write

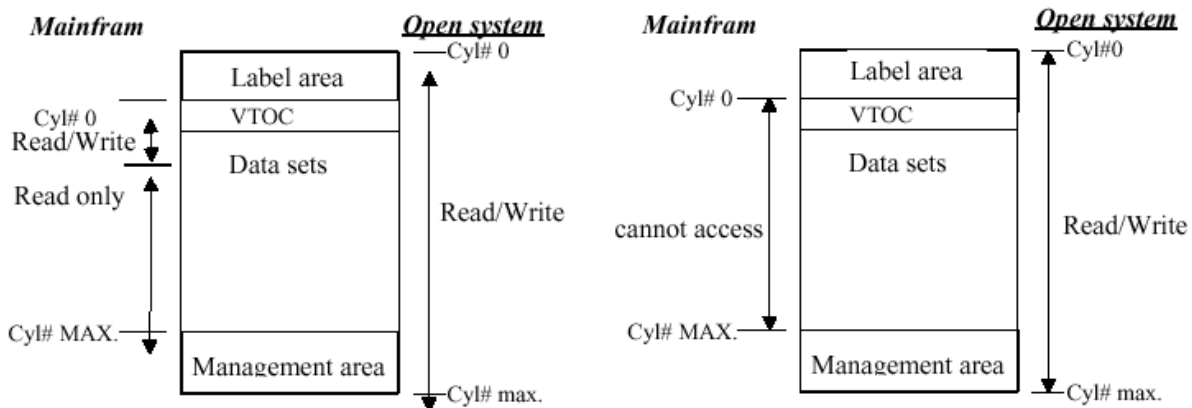
Figure 3 (page 20) and figure 4 (page 20) show the volume configuration of each emulation type



(a) 3390-3B, 3380-KB type

(b) 3390-3A, 3380-KA type

Figure 3. Data Exchange volume configuration



(c) 3390-3C, 3380-KC type

(d) Open-3/8/9/K/E/M (FMT/ALC) type

Figure 4. Data Exchange volume configuration

Data Transfer Directions

FAL/FCU supports the following data transfer directions:

- Data Exchange MTO: from the mainframe to open system
- Data Exchange OTM: from the open system to the mainframe
- Data Exchange OTO: between open system hosts

Caution *To use Data Exchange OTM, you must first process the dataset so that it is registered in VTOC; complete allocation of the extent; and specify the attributes, such as dataset type, record format, block length and record length. Because the extent is not expanded automatically by this program, make sure you have enough to store the data from the open system.*

Caution *To use Data Exchange OTO, you must first format the Open-3 volume (which is only allowed for this operation) as if it is a mainframe volume using FMT. To use ALC, you must first process the dataset in the volume so that it is registered in VTOC; complete allocation of the extent; and specify the attributes, such as dataset type, record format, block length, and record length. Because the extent is not expanded automatically by this program, make sure you have enough to store the data from the open system.*

Using This Manual for Data Exchange OTO

When using this manual for Data Exchange OTO function, be aware of the following notes.

- In the following chapters and sections, files on the Data Exchange volume is expressed as mainframe datasets or mainframe files from the mainframe host side view.
- Read them as open system files when you are using Data Exchange OTO.
- Data Exchange OTO operation is realized as analogy of the combination of Data Exchange MTO and Data Exchange OTM. Assuming that open system host-A and host-C are trying to exchange data from A to C via Data Exchange volume-B, the operation A to B is

just like Data Exchange OTM. And, the operation B to C is also just like Data Exchange MTO.

- The FMT and ALC are included in the FAL/FCU program and must be used to create the mainframe SAM dataset prior to the Data Exchange OTO operation to realize the above analogy.

Pipe Functions

FCU only supports data transfer from the mainframe to the application program or the utility program on a UNIX system using a named pipe. See “Pipe Functions” (page 56).

Code Systems Supported

FCU supports EBCDIC as a code system of the mainframe and ASCII as a code system of the open system. FCU supports an EBCDIC - ASCII code conversion of data entities, if specified so. In addition, you can specify any one byte code conversion table as a file in FCU. See “Code Converting Functions” (page 58).

Caution *Do not use the code conversion function for Data Exchange OTO, or the result of the conversion cannot be guaranteed.*

Server Machines Supported

Table 3. Server Machines Supported

Hardware Platform	OS	GUI ^a	Remarks
SUN	Solaris 2.5, 2.6, 7, 8 ^b	Motif 1.2	FAL/FCU can be used for FC (Fibre Channel) connection as well as SCSI, if the DKC has supported it. (The DKC of XP512 is not a supported SCSI)
HP	HP-UX 10.2, 11.0	Motif 1.2	
IBM	AIX 4.1.4, 4.2, 4.3	Motif 1.2	
(Not specified)	Windows NT 4.0	-	
	Windows 2000		
Compaq	Digital UNIX 4.0	Motif 1.2	
	Tru64 UNIX 4.0, 5.0		
NCR	UNIX SVR4 3.02	Motif 1.2	
Sequent NUMA-Q	DYNIX/ptx 4.4.4	Motif 1.2	

a. GUI window system software is required when FCU is executed using the Operation window.

b. Only version 48.XX.20 or above.

Supported Server Machine Combinations for Data Exchange OTO Operations

Table 4. Server Machine Combination Allowed for Data Exchange OTO Operation

		IBM AIX	HP HP-UX	SEQUENT DYNIX	Windows NT/2000	Digital/ Tru64	SUN Solaris	NCR SVR4
Non-label	IBM AIX	OK	OK	OK	OK	OK	OK	OK
	HP-UX	OK	OK	OK	OK	OK	OK	OK
	SEQUENT DYNIX	OK	OK	OK	OK	OK	OK	OK
Label write option	Windows NT/2000	OK	OK	OK	CHK	OK	CHK	CHK
	Digital/Tru64	OK	OK	OK	OK	CHK	CHK	CHK
Label auto-write	SUN Solaris	OK	OK	OK	OK	OK	NG	NG
	NCR SVR4	OK	OK	OK	OK	OK	NG	NG

Table legend:

- **OK**: Allowed combination
- **CHK**: Not allowed after left servers write a label
- **NG**: Not allowed

FILE ACCESS LIBRARY (FAL) FUNCTIONS

FAL provides following functions in the form of functions in C:

Table 1. FAL Functions

Function	Description
datasetOpen	Opens a specified dataset by a volume name and a dataset name (page 26)
datasetGet	Reads one record from a specified dataset (page 28)
datasetPut/datasetPut2	Writes one record to a specified dataset (page 30)
datasetClose	Closes a specified dataset (page 32)
datasetGetLastError	Acquires error information (page 32)

Table 1. FAL Functions (continued)

Function	Description
datasetGrFileInformation	Acquire dataset attribute information (page 33)
datasetFindFirstFile	
datasetFindNextFile	
datasetFindClose	
datasetGetDsorgString	
datasetGetRecfmString	

- The FAL is not “thread safe” and cannot guarantee the operation when the FAL is used by multiple “threads” in a single process. It is not allowed to use the FAL from “signal handler”.
- The following terminology is reserved for FAL, and cannot be used as function names, variable names or constant symbol names to create a program with FAL:

Words beginning with **dataset**

Words beginning with **fast_**

“GetVolSers

Opening a Dataset

This function opens a dataset (file), specified by pathname with open mode.

[Description format] **datasetHandle = datasetOpen (pathname, mode)**

		Type	Description
Argument:	pathname mode	char* char*	Path name (volume name:dataset name (, VSN identification)) Open mode
Return value:	datasetHandle -1	DATASET_HANDLE	Handle Error end

Argument Details

pathname	Set of a volume name and a dataset name. These names must be separated by a delimiter “.”
Volume name	Six-digit volume serial number (VSN) defined by the mainframe
Dataset name	Dataset name (up to 44 digits long) defined by the mainframe Dataset names cannot contain spaces
VSN identification	Specify the same VSN identification in the volume definition file (you can omit this parameter)
mode	r: Opens a dataset with read access w: Opens a dataset with write access

Return Value Details

This function returns a handle for a dataset specified by the pathname when the file opening terminates successfully.

datasetHandle	Information to identify the dataset opened. The information is used for the argument of subsequent functions, datasetGet , datasetPut , and datasetClose . Do not change the value or the normal functionality cannot be guaranteed.
---------------	---

“-1” is returned when the function terminates unsuccessfully. To get the detailed error information, execute **datasetGetLastError** function described in “Error Information” (page 39).

Notes

- The above volume name relates to a volume (partition) defined by the open system by using “volume definition-file” (file name = datasetmount.dat, covered in “Volume Definition Files” (page 83)).
- Only one dataset can be opened at the same time within one process. When multiple datasets need to be processed, a dataset being opened must be closed once, then another dataset must be opened.
- You cannot open any datasets while acquiring dataset attributes using **datasetFindFirstFile** or **datasetFindNextFile**. You can open datasets after finishing the acquiring dataset attributes using **datasetFindClose**.
- Any files on open system can be opened regardless of restrictions above.
- The datasets which can be processed by this library are limited. Refer to the description in “Supported Objects” (page 16).

Reading Data

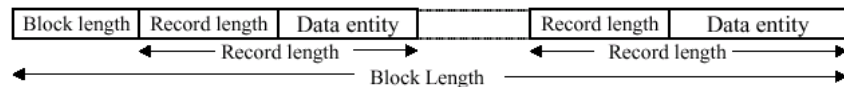
This function reads one record from a dataset specified with a “datasetHandle” and puts it into a “buf” (buffer). Only the actual data entity is transferred to the buffer, as shown below.

[Description format] **reclen = datasetGet (datasetHandle, buf, buflen)**

		Type	Description
Argument:	datasetHandle	DATASET_HANDLE	Dataset handle
	buf	char*	Read buffer
	buflen	long	Buffer size
Return value:	reclen	long	Data entity size transferred to the buffer
	-1		Error end

- Record format = fixed length 
- Record format = variable length 

When a record type is variable, a block and records must be the following format:



- Block length format

← 2 bytes →	← 2 bytes →
Block length	0x0000
- Data length format

← 2 bytes →	← 2 bytes →
Record length	0x0000

Argument Details

<code>datasetHandle</code>	Specifies the handle which is a returned value of <code>datasetOpen</code>
----------------------------	---

buf	Specify the area to store the read data
-----	---

buflen	Specify the size of a provided “buf”. If an actual record size is greater than the “buflen” or equal to zero, this function returns an error without transferring data to the “buf”.
--------	--

Return Value Details

When the data reading terminates successfully, this function returns the length of the read data.

When it detects an end of the dataset (end of file), or when the data reading terminates unsuccessfully, this function returns “-1”. Detailed error information can be obtained by using a **datasetGetLastError** function (to be explained later). For example, when EOF is detected in the above case, DATASET ERROR END OF FILE will be returned.

Writing Data

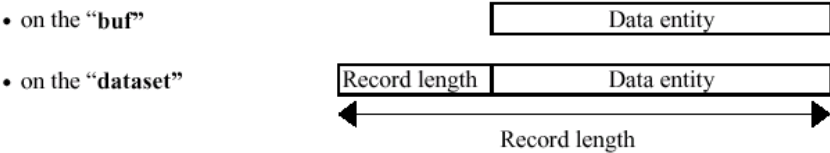
This function writes one record from a “buf” into a dataset specified with a “datasetHandle”.

[Description format] reflen = datasetPut (datasetHandle, buf, buflen)

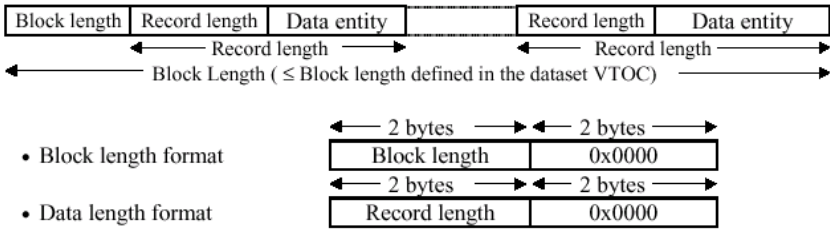
reflen = datasetPut 2(datasetHandle, buf, buflen)

		Type	Description
Argument:	datasetHandle	DATASET_HANDLE	Dataset handle
	buf	char*	Write data
Return value:	buflen	long	Data size
	reflen	long	Data entity size written into the buffer
			Error end

When the record type is variable, the record length is added to the data entity and they are written into the buffer, as shown below.



When it is written into a dataset, multiple records are blocked within the extent defined by the VTOC of the dataset. An example for one which has a variable length of the record type is shown below.



Argument Details

datasetHandle	Specifies a handle which is a returned value of datasetOpen .
buf	Specifies an area where the data to be written is stored.
buflen	<p>Specifies a size of the data. If it is one of the following conditions, an error will be returned:</p> <p>Record type = fixed length: (buflen)?(record length of the dataset)</p> <p>Record type = variable</p> <ul style="list-style-type: none">• (buflen + 4) > (record length of the dataset)• (buflen) = 0 (i.e., no data entity) (When 'datasetPut()' is used an error will be returned, but when 'datasetPut2' is used it will be normal end. 'datasetPut2' supports a record with no data entry.)

Return Value Details

When it terminates successfully, this function returns the length of the written data entity.

When it terminates unsuccessfully, this function returns "-1". Detailed error information can be obtained by using a **datasetGetLastError** function. See the description of **datasetGetLastError** in table 2 (page 40).

Closing a Dataset

This function closes a dataset specified with **datasetHandle**.

[Description format] **datasetError = datasetClose (datasetHandle)**

	Type	Description
Argument: datasetHandle Return value: 0 -1	DATASET_HANDLE	Dataset handle Normal End Abnormal end

Argument Details

datasetHandle Specify the handle which is a returned value of **datasetOpen**

Return Value Details

When the data reading terminates successfully, this function returns “0”.

When it terminates unsuccessfully, this function returns “-1”. Detailed error information can be obtained by using a **datasetGetLastError** function. See the description of **datasetGetLastError** in table 2 (page 40).

Acquiring Error Information

This function acquires detailed information (error code) of the latest error occurred.

[Description format] **datasetLastError = datasetGetLastError()**

	Type	Description
Argument: None Return value: datasetLastError	DATASET_HANDLE	Error code (see Notes)

- Errors specific to the FAL are defined by an error code file (“dataset.h” to be explained later).
- (*UNIX only*) Errors are defined by a standard error file (“errno.h”).
- (*Windows NT only*) Errors are defined by “errno.h” and “winerror.h” attached with Microsoft Visual C++.

Acquiring Dataset Attribute Information

Acquiring Specific Dataset Attribute Information

This function acquires information of attributes of a dataset specified with “pathname” and return it into “ffd”.

[Description format] datasetError = datasetGetFileInformation (pathname, &ffd)

		Type	Description
Argument:	pathname	char*	Path name (VolumeName:DatasetName (, VSN identification))
	ffd	DATASET_FIND_DATA	Attribute information
Return value:	0		Normal end
	-1		Abnormal end

Argument Details

Pathname Indicates the set of a volume name and a dataset name. These names must be separated by a delimiter “:”

Volume name Specifies a six-digit volume serial number (VSN) defined by the mainframe

Dataset name Specifies a dataset name (up to 44 digits long) defined by the mainframe

Dataset names cannot contain spaces

VSN identification Specifies the same VSN identification in the volume definition file (you can omit this parameter)

ffd Specifies the area where the dataset attributes information is stored

The contents of the dataset information are shown below:

```
typedef struct DATASET_FIND_DATA {  
  
    unsigned short blockSize;    /* Block length */  
    unsigned short recordSize;   /* Record length */  
    unsigned char dsorg[2];      /* dataset type */  
    unsigned char recfm; /* record format */  
    char name[44];               /* dataset name */  
    unsigned short lastBlockTt; /* last block address (relative track  
                                number) */  
    unsigned char lastBlockR; /* last block address (relative record  
                                number) */  
} DATASET_FIND_DATA;
```

Return Value Details

When it terminates successfully, this function returns “0”.

When it terminates unsuccessfully, this function returns “-1”. Detailed error information can be obtained by using a **datasetGetLastError** function. See the description of **datasetGetLastError** in table 2 (page 40).

- Notes*
- This function also has the same restriction as in **datasetOpen**. See the description of **datasetOpen** in table 2 (page 40).
 - This function cannot be used while a mainframe file is being opened. Use this function either prior to opening the mainframe file or after closing the file.
 - Do not refer to lastBlockTt when the dataset size is larger than 0xFFFF tracks.

Acquiring Attribute Information of Multiple Datasets

This can be done by using a combination of **datasetFindFirstFile**, **datasetFindNextFile**, and **datasetFindClose**.

Finding the First Dataset Attribute

This function acquires attribute information of the first dataset included in a volume specified by “pathname”, and return it into “ffd”. When the subsequent dataset information is needed, **datasetFindNextFile** can be used for it.

- Notes*
- This function also has the same restriction as in **datasetOpen**. See the description in **datasetOpen** in table 2 (page 40).
 - This function cannot be used while a mainframe file is being opened. Use this function either prior to opening the mainframe file or after closing the file.
 - Mainframe file cannot be opened during the time between this function is called and **datasetFindClose** is called.

[Description format] **datasetHandle** = **datasetFindFirstFile** (pathname, &ffd)

		Type	Description
Argument:	pathname	char*	Path name (VolumeName(, VSN identification))
	ffd	DATASET_FIND_DATA	Attribute information
Return value:	datasetHandle	DATASET_HANDLE	Normal end
	-1		Abnormal end

Argument Details

Pathname	Volume name. Specifies a six-digit volume serial number (VSN) defined by the mainframe
----------	--

VSN identification	Specifies the same VSN identification in the volume definition file (you can omit this parameter)
--------------------	---

The contents of dataset attribute information (**DATASET_FIND_DATA**) are the same as of **datasetGetFileInformation**. See the description of **datasetGetFileInformation** in table 2 (page 40).

Return Value Details When it terminates successfully, the handle relating to the dataset referred the attribute information will be returned. Do not change the contents of the Handle, because it is used for the argument of subsequently used **datasetFindNextFile** and **datasetFindClose**, or the normal functionality cannot be guaranteed.

When it terminates unsuccessfully, this function returns “-1”. Detailed error information can be obtained by using a “datasetGetLastError” function. For example, when no dataset is found in the VTOC, “DATASET_ERROR_END_NO_DATASET” will be returned.

Finding the Next Dataset Attribute

This function is usually used to acquire attribute information of the next dataset in the volume specified by **datasetHandle**. This function should repeat until “no dataset found” is returned or until the user program judges no more information is needed.

- Notes*
- This function also has the same restriction as in **datasetOpen**. See the description in **datasetOpen** in table 2 (page 40).
 - This function cannot be used while a mainframe file is being opened. Use this function either prior to opening the mainframe file or after closing the file.
 - Mainframe file cannot be opened during the time between this function is called and **datasetFindClose** is called.
 - The **datasetFindFirstFile** must be opened prior to using this function.

[Description format] **datasetError = datasetFindNextFile (datasetHandle, &ffd)**

		Type	Description
Argument:	datasetHandle	DATASET_HANDLE	Dataset Handle
	ffd	DATASET_FIND_DATA	Attribute information
Return value:	0		Normal end
	-1		Abnormal end or no dataset found

Argument Details datasetHandle Specify the Handle which was returned in the preceding **datasetFindFirstFile**.

ffd Specify the area where the dataset attributes information is stored.

The contents of dataset attribute information are the same as of **datasetGetFileInformation**. See the description of **datasetGetFileInformation** in table 2 (page 40).

Return Value Details When it terminates successfully, this function returns “0”.

When it terminates unsuccessfully, this function returns “-1”. Detailed error information can be obtained by using a “datasetGetLastError” function. For example, when the next dataset is not found in the VTOC, “DATASET_ERROR_END_OF_VTOC” will be returned.

Finding the End Acquiring Dataset Attribute

This function informs of the termination of acquiring attribute information done by **datasetFindFirstFile** and **datasetFindNextFile**.

- Notes*
- This function must be preceded by **datasetFindFirstFile**.
 - The datasets can be opened after executing this function.

[Description format] datasetError = datasetFindClose (datasetHandle)

		Type	Description
Argument:	datasetHandle	DATASET_HANDLE	Dataset Handle
Return value:	0		Normal end
	-1		Abnormal end

Argument Details datasetHandle Specify the Handle which was returned in the preceding **datasetFindFirstFile**.

Return Value Details When it terminates successfully, this function returns “0”.

When it terminates unsuccessfully, this function returns “-1”. Detailed error information can be obtained by using a **datasetGetLastError** function.

Converting Dataset Organization Types or Record Formats to Character Strings

Converting Dataset Organization Type Values

This function exchanges the value which indicates dataset organization type specified by “dsorg” into a character string text. This function sets the value of “dsorg [], a member of DATASET_FIND_DATA, which is obtained by “datasetGetFileInformation/datasetFindFirstFile/datasetFindNextFile”, to “dsorg”.

[Description format] datasetError = datasetGetDsorgString (dsorg, text)

		Type	Description
Argument:	dsorg text	u_char [] char [3]	Dataset organization type value (2 bytes) Dataset organization type character string (3 bytes) “PS”: Physical Sequential organization “VS”: VSAM organization “DA”: Direct Access organization “PO”: Partial organization “***”: Other than above
Return value:	0 -1		Normal end Abnormal end

Converting Record Format Values

This function exchanges the value which indicates record format specified by “recfm” into a character string text. This function sets the value of recfm, a member of DATASET_FIND_DATA, which is obtained by “datasetGetFileInformation/ datasetFindFirstFile/ datasetFindNextFile”, to “recfm”.

[Description format] datasetError = datasetGetRecfmString (recfm, text)

		Type	Description
Argument:	recfm text	u_char [] char [5]	Record type value (1 byte) Record type character string (5 bytes) text[0]: “F” : Fixed length record “V” : Variable length record “U” : Unknown length record text[1]: “B” : Blocking record “sp” : Spanned record “st” : Standard format record
Return value:	0 -1		Normal end Abnormal end

Error Information

Error information returned by using the **datasetGetLastError** function includes error information (defined by the “dataset.h” file) specific to the FAL.

- Error codes having “*” may be reported when access contention between mainframe and open system on the same volume occurs. If the cause of the error cannot be identified from the contents in the following error message list, please check if any illegal contention on the shared volume occurs.

- An error code with plus value “+” means a system error.

(*UNIX only*) The system error code is defined by the standard error file, “errno.h”, for UNIX system.

(*Windows NT only*) The system error code is defined in the “errno.h” and “winerror.h” attached with Microsoft Visual C++.

Table 2. Error code and error message from File Access Library

Error Code	Error Name	Description	Solution
-2	DATASET_ERROR_ABORTED_BY_SIGNAL	‘Abort’ signal handler is received.	Do not push Ctrl+C and Ctrl+Z, when FAL is executing.
-6	DATASET_ERROR_VOLUME_NOT_MOUNTED	The volume is not mounted.	Retry after the volume is mounted
-7	DATASET_ERROR_INVALID_VOLUME	The VSN on the volume definition file is different from one of the actual volume.	Check the VSN in the volume and one in the volume definition file.
-8	DATASET_ERROR_DATASET_NOT_FOUND	The target dataset is not found.	Check if the actual dataset name and specified dataset name is the same, by using “MF-File list” in HELP menu or by using VTOC dump data in the mainframe side.
-9	DATASET_ERROR_NOT_SUPPORTED	The data format used is not supported.	Check if the dataset is correctly created by mainframe side.
-10*	DATASET_ERROR_DEVICE_TYPE_NOT_SUPPORTED	The device emulation type used is not supported.	Check if the device emulation type is set to the supported one in the volume definition file.
-11	DATASET_ERROR_DSORG_NOT_SUPPORTED	The dataset organization type used is not supported.	Check the dataset organization type by “MF-File list” in HELP menu in File Conversion Utility or by VTOC dump data in the mainframe side

Table 2. Error code and error message from File Access Library (continued)

Error Code	Error Name	Description	Solution
-12	DATASET_ERROR_RECFCM_NOT_SUPPORTED	The record format used is not supported.	Check the dataset organization type by “MF-File list” in HELP menu in File Conversion Utility or by VTOC dump data in the mainframe side.
-13*	DATASET_ERROR_INVALID_DATA	The data in VTOC or the dataset is invalid.	Check if the VTOC and dataset are correctly created by mainframe side.
-14*	DATASET_ERROR_VOLUME_DEFINITION_INVALID	The format of volume definition file is invalid.	Check if the volume definition file is correctly created as defined in the manuals.
-15	DATASET_ERROR_DATASET_NOT_OPENED	An attempt was made to read the dataset without opening it.	Check the program if the “datasetOpen” is called to the dataset before calling “datasetGet”.

Table 2. Error code and error message from File Access Library (continued)

Error Code	Error Name	Description	Solution
-16	DATASET_ERROR_DATASET_NOT_CLOSED	An attempt was made to open the dataset without closing it.	<p>Check if the executed programs conform with the following items;</p> <p>a) Dataset open and close must be used as a pair.</p> <p>b) More than one datasets cannot be opened within one process.</p> <p>c) The “datasetOpen”, “datasetGetFileInformation” or “datasetFindFirstFile” cannot be used during the process of obtaining dataset attribute information by “datasetGetFirstFile” or “datasetFindNextFile”.</p> <p>d) “DatasetGetFileInformation” or “DatasetFindFirstFile” cannot be used for a dataset which is being opened by “DatasetOpen”.</p>
-17	DATASET_ERROR_BUFLEN_SHORT	The buffer length specified by datasetGet is shorter than the record length.	Check the program if the buffer area larger than the dataset record length is prepared.

Table 2. Error code and error message from File Access Library (continued)

Error Code	Error Name	Description	Solution
-18*	DATASET_ERROR_VOLUME_LABEL_INVALID	No standard volume label is found or contents of VTOC are illegal.	Check if the volume initialization is completed correctly from the mainframe side. This error is occurred when the system that is not supported Large File accesses the Volume formatted using the system that is supported Large File. This error is occurred when a data partition size is not right in case of SUN/Solaris.
-19*	DATASET_ERROR_VTOC_INVALID	No VTOC is found or contents of VTOC are invalid.	Check if the VTOC is correctly created by the mainframe side.
-20*	DATASET_ERROR_VOLUME_NOT_DEFINED	The specified volume is not defined.	Check the contents of the volume definition file.
-21	DATASET_ERROR_INVALID_ARGUMENT	An argument of the function is invalid.	Check if the argument specified to the function is correct.
-22	DATASET_ERROR_NO_DATASET	No dataset is found.	Check if the dataset is created by the mainframe side.
-23*	DATASET_ERROR_NON_STANDARD_R0_EXIST	Non standard R0 exists.	Change the track format to standard track format. FAL cannot write data on the track where non-standard R0 is existing.
-24	DATASET_ERROR_INVALID_MODE	Mode argument of datasetOpen is wrong.	Check the mode argument. Only the value of “r” or “w” is allowed.

Table 2. Error code and error message from File Access Library (continued)

Error Code	Error Name	Description	Solution
-26	FAL_INTERNAL_ERROR	Internal error of FAL	Collect error logs file and error dump file. And make contact a maintenance staff. This error is occurred when open system does not have enough memory.
-25*	DATASET_ERROR_VOLUME_DEFINED_READ_ONLY	Open system tried to use read only volume.	Check the volume emulation type. Only the emulation type of xxxx-XAorXC is allowed for open system to write data.
-27*	DATASET_ERROR_END_OF_VOLUME	The end of volume is detected before detecting the end of dataset.	The volume (partition) size defined in open system is set smaller than that recognized in mainframe. Check if the partition size of the open system is correct. This error is occurred when the open system disk is full or it exceeds a limitation in The case of Data Exchange MTO.
-28	DATASET_ERROR_OVERFLOW	Data cannot be written because the dataset is full.	Check the size of data written into dataset, and extend area size of the dataset.
-33	DATASET_ERROR_PARAMETER_MISMATCH	Record format, Record length, or Block length does not match with ones defined in VTOC. Record format, Record length, or Block length is not defined in VTOC nor specified in the parameter.	Check and set the correct parameters.

Table 2. Error code and error message from File Access Library (continued)

Error Code	Error Name	Description	Solution
-35	DATASET_ERROR_NO_LICENSE	FAL cannot permit execution that does not have a software license.	
-36	DATASET_ERROR_TIME_OUT_LICENSE	FAL cannot permit execution that had an expired trying time limit of a software license.	
-37	DATASET_ERROR_HOSTNAME_CHANGE	FAL cannot permit execution that this host and a host installed are not identical and/or hostname is changed.	
-39	DATASET_ERROR_MULTI_VOLUME_DEFINITION_RECORD_OVER	The number of parameter sets for multiple volume definition file exceeded 1000.	Parameter sets more than 1000 cannot be processed. Decrease them not to exceed 1000.
-40	DATASET_ERROR_MULTI_VOLUME_NO_DATASET	The data set is not exist in the next volume.	Check volume serial number in the multiple volume definition file.
-41	DATASET_ERROR_MULTI_VOLUME_NO_TRANSFER	Data cannot be transferred to the dataset that is in middle volume of multiple volume.	
-42	DATASET_ERROR_MULTI_VOLUME_DEFINITION_INVALID_RECORD_LENGTH	The record length in the multiple volume definition file is too long.	Specify the record length less than 1400 characters.(not include delimiter)
-43	DATASET_ERROR_MULTI_VOLUME_DEFINITION_PARAMETER_ERROR	The number of volume for one dataset in the multi volume definition file exceeded 31.	Specify the number of volume is less than 31 for one line in the multi volume definition file.
-44	DATASET_ERROR_MULTI_VOLUME_DEFINITION_NO_DATASET	The Dataset name is not specified in the multiple volume definition file.	specified the dataset name in the head volume information of the multiple volume definition file.

Table 2. Error code and error message from File Access Library (continued)

Error Code	Error Name	Description	Solution
-45	DATASET_ERROR_MULTI_VOLUME_DEFINITION_VSN_LENGTH_ERROR	VSN is incorrect in the multiple volume definition file.	Check if VSN length in the multiple definition file is less than 7.
-46	DATASET_ERROR_MULTI_VOLUME_DEFINITION_DSN_LENGTH_ERROR	DSN is incorrect in the multiple volume definition file.	Check if DSN length in the multiple definition file is less than 45.
-47	DATASET_ERROR_MULTI_VOLUME_DEFINITION_VOLID_LENGTH_ERROR	The VSN identification length in the multiple volume definition file is too long.	Specify the VSN identification length less than 36 characters.
-48	DATASET_ERROR_MULTI_VOLUME_DEFINITION_NO_NEXT_VOLUME	The next VSN is specified in the multiple volume definition file when the volume is not last on VTOC.	Specified all volumes in the multiple volume definition file.
-50*	DATASET_ERROR_END_OF_FILE	End of File (EOF) was detected.	
-51*	DATASET_ERROR_END_OF_VTOC	End of VTOC was detected.	

HOW TO USE FILE ACCESS LIBRARY (FAL)

This chapter covers the following topics for the FAL operational environment and program:

- “Shared Volume Definition” (page 48)
- “Creating a Volume Definition File” (page 48)
- “Installing File Access Library” (page 48)
- “Compiling File Access Library” (page 48)

Providing Operational Environment

The operational environment includes shared volumes, volume definition files, and using File Access Library (FAL).

Shared Volume Definition

A volume for storing a dataset to be accessed by FAL must be defined as a shared volume. See “Supported Objects” (page 16) in this manual for the types of shared volumes supported.

Creating a Volume Definition File

The library requires a volume definition file to relate logical volumes which are defined by the mainframe to partitions which are defined by the open system. Create the volume definition file in the same directory where the application program using the library is registered. See “Volume Definition Files” (page 83) for details about volume definition files.

Installing File Access Library

- Install from the media where the FAL is stored according to the procedure described in “Providing Operational Environment” (page 70) in this manual.
- Include the header file within the application program that uses the FAL.

Compiling File Access Library

The following examples show how to build FAL in a user’s program. For UNIX systems, use the C compiler that conforms with ANSI. This library does not support C++.

AIX

cc -qlanglvl=ansi -o Output file name Source file name /usr/lib/libfal.a

- libfal.a: This specifies the file name of an object module in the File Access Library.

HP-UX

cc -Ae +DAportable -o Output file name Source file name /usr/lib/libfal.sl

- libfal.sl: This specifies the file name of an object module in the File Access Library.

Solaris

cc -o Output file name Source file name /usr/lib/libfal.so.1

- libfal.so.1: This specifies the file name of an object module in the File Access Library.

Digital UNIX/Tru64 UNIX

c89 -o Output file name Source file name /usr/lib/libfal.so

- libfal.so: This specifies the file name of an object module in the File Access Library.

NCR UNIX

cc -o Output file name Source file name /usr/lib/libfal.so

- libfal.so: This specifies the file name of an object module in the File Access Library.

DYNIX/ptx

cc -lseq -o Output file name Source file name /usr/lib/libfal.so

- libfal.so: This specifies the file name of an object module in the File Access Library.

Windows NT

1. Start Developer Studio.
2. Create a new project.
3. Copy the following three FAL files into a folder.

dataset.h

fal.dll

fal.lib

- In Developer Studio, click Projects | Settings.
4. In the Project Settings dialog box, click the Link tab.
 5. Add “**fal.lib**” into “**object/library module**” field.
 6. Build and execute.

Using the File Access Library Application Program

Example: Reading Data

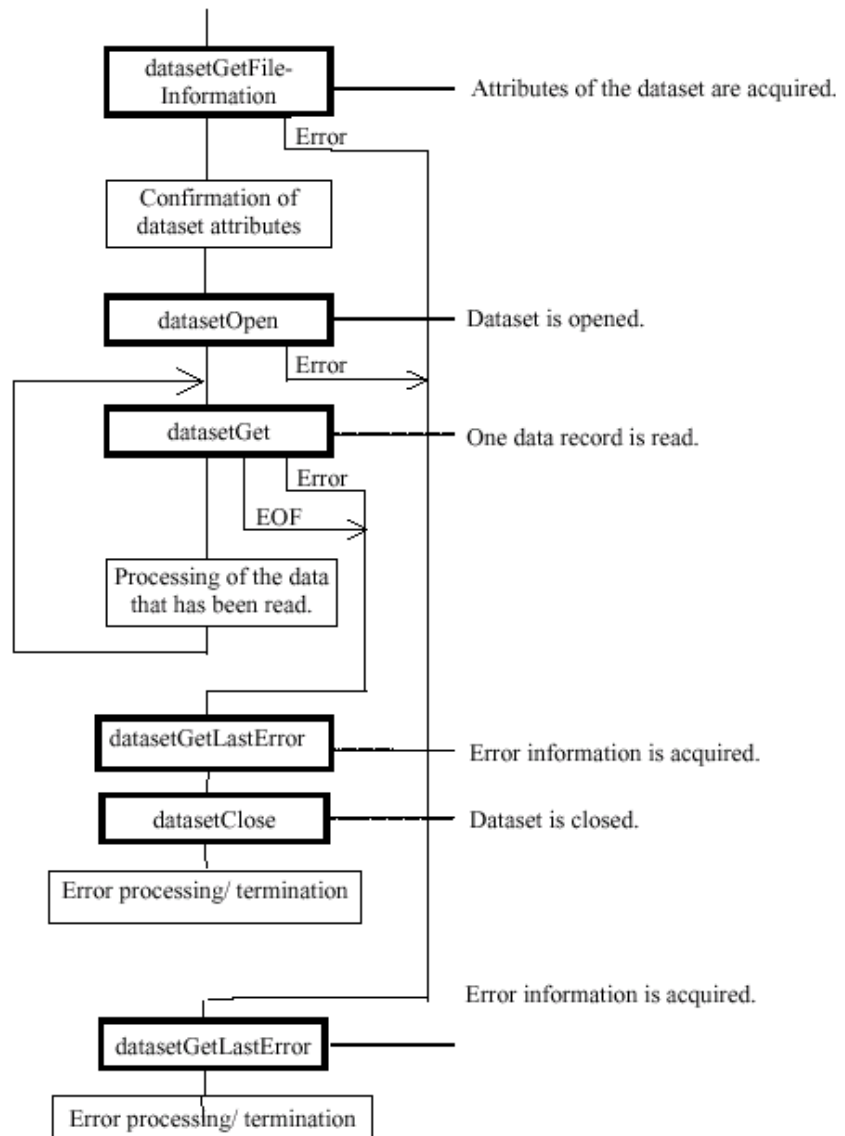


Figure 1. `datasetGetFileInformation`

Example: Acquiring Dataset Attributes

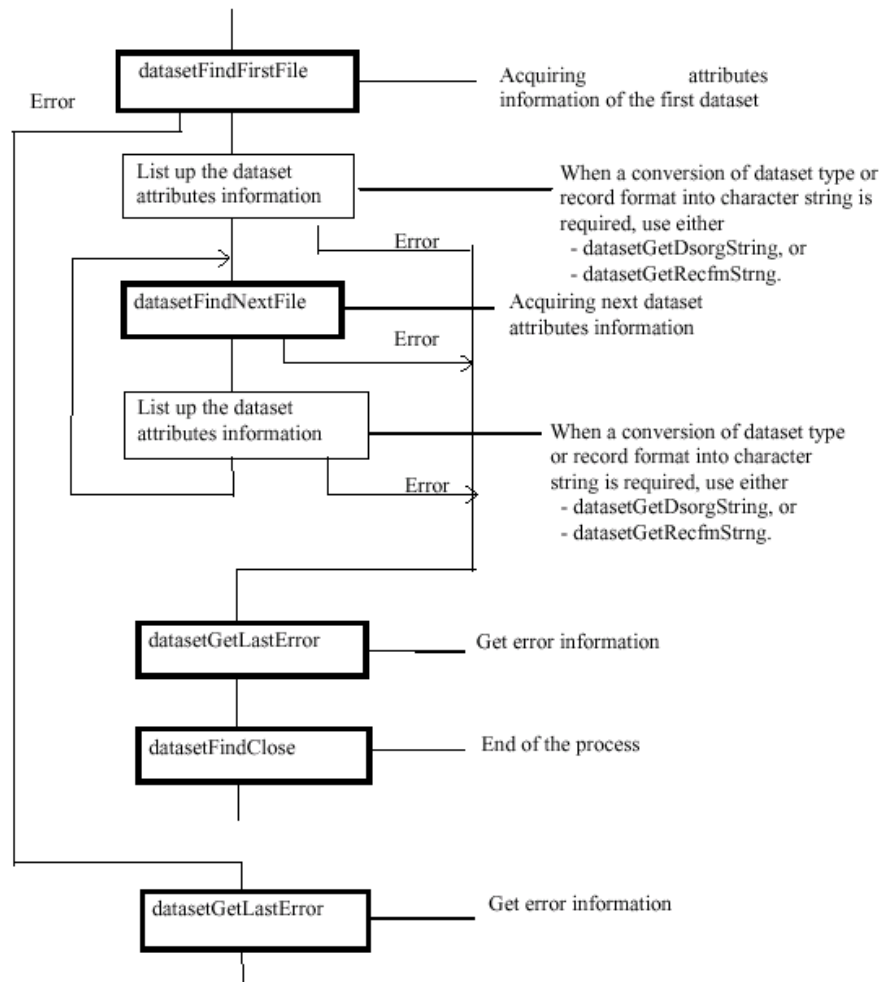


Figure 2. *datasetFindFirstFile*

FUNCTIONS

This chapter describes the following supporting functions:

- “File Conversion Functions” (page 54)
- “Pipe Functions” (page 56)
- “Code Converting Functions” (page 58)
- “Padding Functions” (page 60)
- “Delimiter Adding Functions” (page 62)
- “Record Length Adding Function” (page 66)
- “Data Exchange from Multiple Volume Datasets” (page 67)
- “Bi-directional Data Transfer between Mainframe and Open Systems” (page 68)

File Conversion Functions

This function transfers only data entities between a mainframe dataset and an open system file specified. Object data entities are the ones contained in all records between the beginning of a file and the end of a file.

Note: End of file

Mainframe dataset: EOF record or end of the final extent

Open system file: EOF

Data Editing Formats

Fixed Length Dataset Record Types

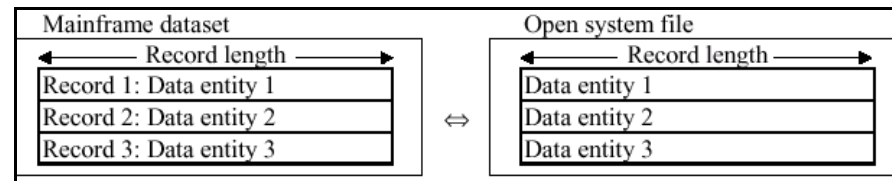


Figure 1. Mainframe dataset open system file fixed length

- Record length is one defined as an attribute of the dataset, and equal to the length of each data entity.

Variable Length Dataset Record Types

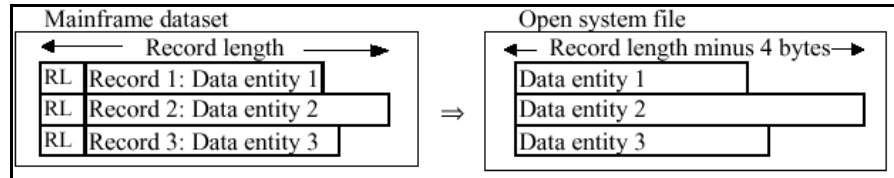


Figure 2. Mainframe dataset open system file variable length]

- Record length is one defined as an attribute of the dataset, and shows maximum record length.
- RL represents the length of each data entity + 4 bytes in each record. RL is not transferred to the open system file when transferred from mainframe to open system.

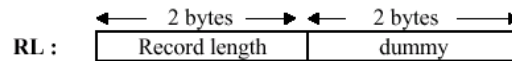


Figure 3. Record length/dummy

- When data is transferred from open system to mainframe and the delimiter (described later) is not specified, the maximum record length is used for all records to execute code conversion. Thus, if variable length records are processed without delimiter, data of each record is destroyed.

Pipe Functions

This function transfers data entries from the mainframe to the application program or the utility program in UNIX system using a named pipe, when pipe function is specified. When this function is used, mainframe dataset can be transferred to open system, higher-speed than the way of outputting and convert a file once.

Caution *This function is only supported in UNIX system (Not supported in Windows NT system). Only this function for data transfer from the mainframe to the open system is supported. This function is necessary an application program or a utility program to receive data entries using a named pipe.*

Time Out Value Definition

When transfer data entries use a named pipe, FCU waits for the status that it can be written in a named pipe (the status that an application program or a utility program can read data entries). If the status does not become available before the defined time out period, a time out error is reported.

The definition of time out value can be setting on [WAIT_TIME_VALUE] of the environment variable.

Limits: 0~1440 minutes (0 means unlimited)

Default: 0 (when not setting the definition of the time out value)

The environment variable are defined with the following examples. When complete the setting below, log out and log in again.

When using C shell, add the following line to the file **.cshrc** in the home directory. If **.cshrc** does not exist, create it and describe it as explained here.

Setenv WAIT_TIME_VALUE 300

When using other than C shell, add the following two lines to the file **.dtprofile** in the home directory. If **.dtprofile** does not exist, create it and describe it as explained here.

WAIT_TIME_VALUE=300

Export WAIT_TIME_VALUE

Add the two lines to the file **.profile** in the home directory when it is not the common desktop environment. If **.profile** does not exist, create it.

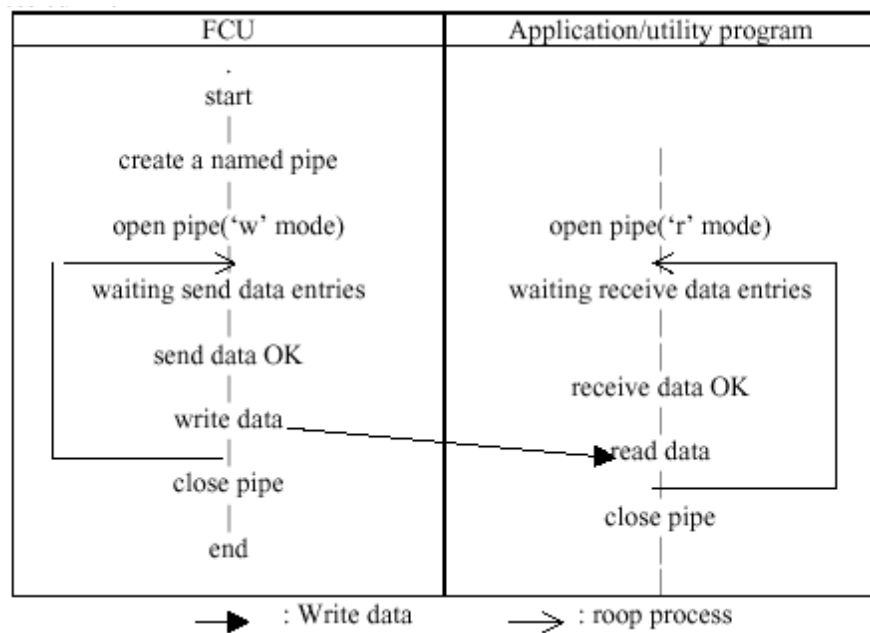


Figure 4. Process outline

Code Converting Functions

When code conversion is specified, this function performs code conversion of data entities. Two kinds of conversion method is supported. One is to use the conversion table which is included in FCU. The other is a user to provide the conversion table and to specify the file name as a parameter.

Using the Code Conversion Table Included in FCU

Only a conversion from EBCDIC to ASCII for data transfer from mainframe to open system, and from ASCII to EBCDIC for that from open to mainframe host is supported.

Using the User-defined Code Conversion Table

You can create your own code conversion table as a file and specify the file name as a FCU parameter. Refer to section “Creating New Parameter Definition Files” (page 128) and “Parameter Definition Files” (page 95) about how to specify the code conversion table file.

- The code conversion table must be 256 bytes in size and must be a binary data.
- Only one byte conversion is supported. Two bytes code cannot be converted.
- The code conversion table is created as putting a target code into the offset position in the table which represents the code value of the source code to be converted.

Table 1. Code conversion between EBCDIC and ASCII

H L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL (00)	DLE (10)	DS (80)	(90)	SP (20)	& (26)	- (2D)	(BA)	(C3)	(CA)	(D1)	(D8)	{ (7B)	} (7D)	\ (5C)	0 (30)
1	SOH (01)	DC1 (11)	SOS (81)	(91)	(A0)	(A9)	/ (2F)	(BB)	a (61)	j (6A)	(E5)	(D9)	A (41)	J (4A)	(9F)	1 (31)
2	STX (02)	DC2 (12)	FS (82)	SYN (16)	(A1)	(AA)	(B2)	(BC)	b (62)	k (6B)	s (73)	(DA)	B (42)	K (4B)	S (53)	2 (32)
3	ETX (03)	DC3 (13)	(83)	(93)	(A2)	(AB)	(B3)	(BD)	c (63)	l (6C)	t (74)	(DB)	C (43)	L (4C)	T (54)	3 (33)
4	PF (9C)	TM	BYP (84)	PN (94)	(A3)	(AC)	(B4)	(BE)	d (64)	m (6D)	u (75)	(DC)	D (44)	M (4D)	U (55)	4 (34)
5	HT (09)	(85)	LF (0A)	RS (95)	(A4)	(AD)	(B5)	(BF)	e (65)	n (6E)	v (76)	(DD)	E (45)	N (4E)	V (56)	5 (35)
6	LC (86)	BS (08)	ETB (17)	UC (96)	(A5)	(AE)	(B5)	(C0)	f (66)	o (6F)	w (77)	(DE)	F (46)	O (4F)	W (57)	6 (36)
7	DEL	IL (87)	ESC (1B)	EOT (04)	(A6)	(AF)	(B7)	(C1)	g (67)	p (70)	x (78)	(DF)	G (47)	P (50)	X (58)	7 (37)
8	GE	CAN (18)	(88)	(98)	(A7)	(B0)	(B8)	(C2)	h (68)	q (71)	y (79)	(E0)	H (48)	Q (51)	Y (59)	8 (38)
9	RLF	EM (19)	(89)	(99)	(A8)	(B1)	(B9)	(60)	i (69)	r (72)	z (7A)	(E1)	I (49)	R (52)	Z (5A)	9 (39)
A	SMM	CC (92)	SW (8A)	(9A)	(D5)	! (21)	(CB)	(3A)	(C4)	^ (5E)	(D2)	(E2)	(E8)	(EE)	(F4)	(FA)
B	VT	CUI (8F)	CUI (8B)	CU3 (9B)	(2E)	\$ (24)	(2C)	# (23)	(C5)	(CC)	(D3)	(E3)	(E9)	(EF)	(F5)	(FB)
C	FF	IFS (1C)	(8C)	DC4 (14)	< (3C)	* (2A)	% (25)	@ (40)	(C6)	(CD)	(D4)	(E4)	(EA)	(F0)	(F6)	(FC)
D	CR	IGS (1D)	ENQ (05)	NAK (15)	((28)) (29)	_ (5F)	' (27)	(C7)	(CE)	[(5B)] (5D)	(EB)	(F1)	(F7)	(FD)
E	SO	IRS (1E)	ACK (06)	(9E)	+ (2B)	; (3B)	> (3E)	= (3D)	(C8)	(CF)	(D6)	(E6)	(EC)	(F2)	(F8)	(FE)
F	SI	IUS (1F)	BEL (07)	SUB (1A)	 (7C)	~ (7E)	? (3F)	" (22)	(C9)	(D0)	(D7)	(E7)	(ED)	(F3)	(F9)	(FF)

Padding Functions

When the record type of a mainframe is variable length and the data is transferred from mainframe to open system, this function pads shorter data entities so that all data entities of a created open system file have the same length (the maximum data length defined for the mainframe dataset).

When the record type of a mainframe is fixed length and the data is transferred from open system to mainframe, this function pads shorter data entities so that all data entities of a created mainframe dataset have the same length (the data length defined for the mainframe dataset).

- The open system file, to which the padding is applied, cannot be restored to the mainframe dataset
- The mainframe dataset, to which the padding is applied, cannot be restored to the open system file
- When the data transfer direction is from open system to mainframe, the delimiter must be set at the end of each open system file (refer to “Delimiter Adding Functions” (page 62) for the delimiter)

This function pads the following data to a dummy area:

- Pads “0x00” when the code conversion is not specified.

The file is assumed to be a binary format.

- Pads “space” when the code conversion is specified.

The file is assumed to be a text format file.

Variable Length MTO Data Transfer

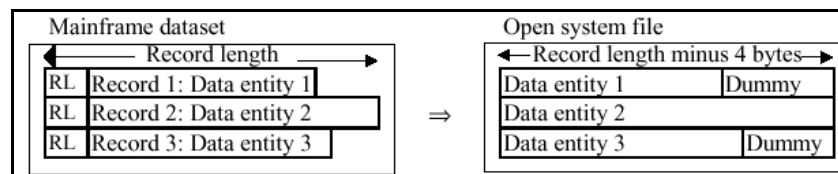


Figure 5. Variable Length MTO Data Transfer

Fixed Length OTM Data Transfer

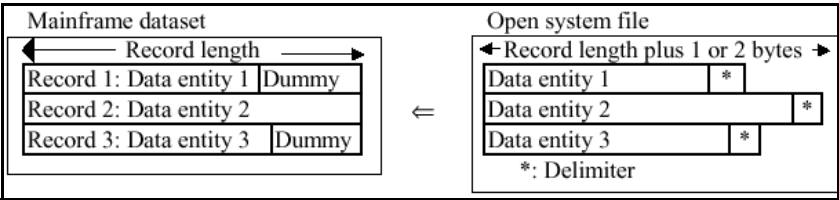


Figure 6. Fixed Length OTM Data Transfer

- Record length is one defined as an attribute of the dataset, and shows maximum record length for the dataset with variable length record format. For the dataset with fixed length, all the records have the same length.
- RL represents the length of each data entity + 4 bytes in each record. RL is not transferred to the open system file when transferred from mainframe to open system.
- The information indicating the block length will not be transferred.

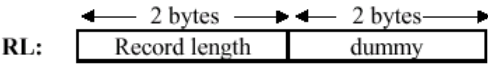


Figure 7. Example of data editing format by padding

Delimiter Adding Functions

This function places a delimiter which is used for separating each data in the open system files. The different characters are used for the delimiter depending on the open system used.

UNIX: CR (carriage return) or LF (line feed)

Windows NT: CR plus LF

Data Editing Formats

Fixed Length Dataset Record Types

In case of data transfer from mainframe to open system, the delimiter is added right after data entity and the data is stored in the open system file.

In case of data transfer from open system to mainframe and the padding is not specified, the data entity is extracted in accordance with the specified record length for the dataset and written into mainframe dataset. If padding is specified, refer to “Padding Functions” (page 60).

- The delimiter is not transferred
- An error is reported when a delimiter is not found right after the record which is extracted by the record length

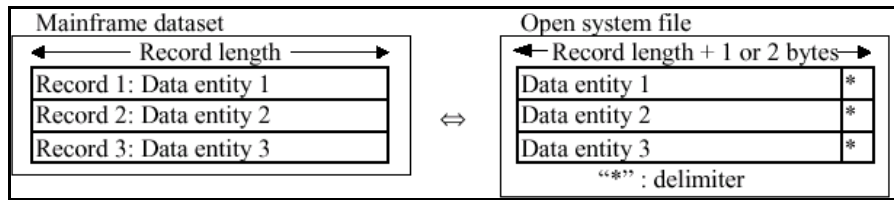


Figure 8. Mainframe dataset open system file

- Record length is one defined as an attribute of the dataset, and equal to one of data entity
- Data length of open system files is as follows:
 UNIX: Record length + 1
 Windows NT: Record length + 2

Variable-length Mainframe Dataset Record Types

For variable-length mainframe dataset record types, you can choose to specify a padding delimiter.

No Padding Specified

For MTO, the delimiter is added right after data entity and the data is stored in the open system file.

For OTM, only data entity right before the delimiter is extracted and written into mainframe dataset.

- The delimiter is not transferred.
- An error is reported when the length up to the delimiter in the data entity is longer than the mainframe record length minus 4.
- “RL” is added when the data is stored into the mainframe dataset.

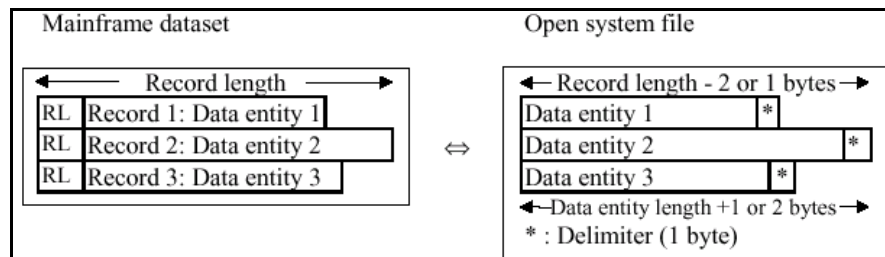


Figure 9. Mainframe dataset pen system file

- Record length is one defined as an attribute of the dataset, and equal to one of data entity.
- Data length of open system files is as follows:
UNIX: data entity length + 1
Windows NT: data entity length + 2
- RL represents the length of each data entity + 4 bytes in each record. RL is not transferred to the open system file when transferred from mainframe to open system.

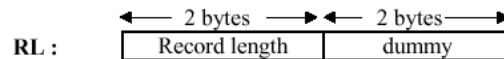


Figure 10. Record length/dummy

Padding Specified

For MTO, the delimiter is added right after data entity.

- When a open system file, to which padding is applied, is returned to mainframe, the dataset does not hold the same data format as the original one, because the dummy area is not removed.

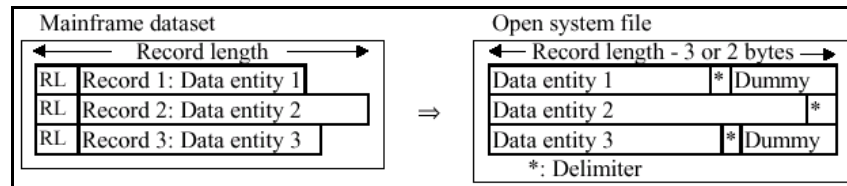


Figure 11. Mainframe dataset open system file

- Record length is one defined as an attribute of the dataset, and representing the maximum length of record
- Data length of open system files is as follows:
 UNIX: record length - 3
 Windows NT: record length - 2
- RL represents the length of each data entity + 4 bytes in each record. RL is not transferred to the open system file when transferred from mainframe to open system

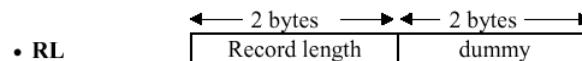


Figure 12. Record length/dummy

Record Length Adding Function

This function creates an open system file in which a record length is added to each data entity when the data transfer direction is mainframe to open system and variable length record format is specified.

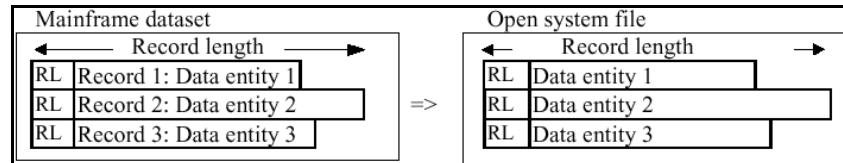


Figure 13. Mainframe dataset open system file

- Record length is one defined as an attribute of the dataset, and representing the maximum length of record.
- RL represents the length of each data entity + 4 bytes in each record.

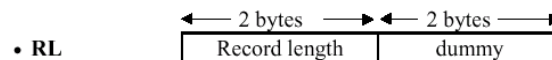


Figure 14. Record length/dummy

Data Exchange from Multiple Volume Datasets

Multiple volume datasets are supported only by Data Exchange MTO, for versions 50.XX.YY (XX=01 or 02) and above.

A multiple volume definition file (**multidef.dat**) must exist in the current directory. See “Multiple Volume Definition Files” (page 93) for multiple volume definition file formats.

FAL came to check dataset serial number and last volume containing data in this dataset in dataset indicators on VTOC DSCB1. (See the following the check spec).

VTOC DSCB1 Action	Dataset serial number			
	1	1	Except 1	Except 1
	Last volume containing data in this dataset in dataset indicators.			
	ON	OFF	ON	OFF
OTM	OK(OK)	OK(OK)	NG(OK)	NG(OK)
MTO for single volume (*1)	OK(OK)	NG(OK)	NG(OK)	NG(OK)

(*1) The dataset does not exist in the multiple volume definition file

The inside of “()”: The spec of i) or ii)

i) Before version 50.XX.YY (XX=01 or 02)

ii) Version is 50.XX.YY or above, and FAL_MULTI_CHECK of the environment variable is off

Bi-directional Data Transfer between Mainframe and Open Systems

1. When transferring mainframe dataset with variable length record format from mainframe to open system, and sending the file back to mainframe to re-use it, use the “Delimiter adding function”. Otherwise, the data will not be compatible with the original one.
2. When a dataset is used in which the same code as a delimiter is included for a different purpose, do not use “Delimiter adding function”. It is no problem to use the delimiter adding function for mainframe to open system transfer direction. However, for open system to mainframe transfer, a character which is not used for a delimiter in the original data will be taken as a delimiter. This will create dataset that is not intended, contents or an error condition in FCU process.
3. When a file is transferred from mainframe to open system with data padding specified and sent it back to mainframe, the data will be incompatible with the original one. This is because padding data cannot be deleted in transferring from open system to mainframe.
4. You cannot use the record length adding function together with the code conversion, delimiter adding, or padding functions. All the parameters specifying code conversion, delimiter adding, and padding must be set to no in this case. In addition, you cannot use the record length adding function when transferring data from open system to mainframe. Therefore, if a file in which record length is added is returned to mainframe, the dataset will be incompatible with original dataset.

HOW TO USE FILE CONVERSION UTILITY (FCU)

This chapter covers the following topics for the FCU operational environment and program, and for definition files:

- “Shared Volume Definition” (page 70)
- “Installing the FAL/FCU Program for UNIX” (page 72)
- “Installing the FAL/FCU Program for Windows NT” (page 79)
- “Inputting License Keys” (page 81)
- “Setting License Keys” (page 82)
- “Volume Definition Files” (page 83)
- “Multiple Volume Definition Files” (page 93)
- “Parameter Definition Files” (page 95)
- “Starting FCU for UNIX” (page 105)
- “Starting FCU for Windows NT” (page 107)

Providing Operational Environment

Shared Volume Definition

A volume for storing a file to be accessed by FCU via the File Access Library (FAL) must be defined as a Data Exchange volume. The Data Exchange volume can be accessed by both the mainframe and the open system. The emulation type supported for the Data Exchange volume is defined in the section “Mainframe Device Emulation Types” (page 19). Refer to the Maintenance Manual for the installation procedure of these volumes.

Notes

- Refer to *XP256/XP512 SCSI Installation Manual* for each platform when partitions on Data Exchange volumes are set by open system servers.
- Set the IO time out value and IO queue depth value to each Data Exchange volume from open system server. The open system server may detect an error if these values are not set correctly. Refer to *XP256/XP512 SCSI Installation Manual* for the operation detail.

- (SUN/Solaris only) You need to specify a data partition size for open systems.

Use the following table to determine number/volume type as a data cylinder extent relationships. In the setting procedure, add alternate cylinder #2.

Volume type	Cylinder # for a data cylinder extent
3390-3A	0 - 3345
3390-3B	0 - 3339
3390-3C	0 - 3345
3380-KA	0 - 2661
3380-KB	0 - 2655
3380-KC	0 - 2661
OPEN-3/8/9/K/E	0 - primary cylinders of the formatter+4
OPEN-L/M	Geometry is special value. Refer to the <i>Solaris Configuration Guide</i> .

Refer to the Installation section in *XP256/XP512 SCSI Installation Manual* for the detailed procedure.

- (SUN/Solaris only) You can ignore the following warning messages for open systems created during label writing in the volume formatting process, and continue the operation.

Warning: error writing VTOC

Warning: no backup labels

Write label failed

- Do not share the same Data Exchange volume between Solaris and NCR UNIX, because Solaris and NCR UNIX always write a label on the volume, and Windows NT/2000 and DIGITAL/Tru64 UNIX have an option to write a label on the volume. Additionally, after both connecting servers write a label on the Data Exchange volume, the Data Exchange volume cannot be shared between servers.
- Access privilege to shared volumes is given only to **root**. Specify the access rights to the shared volumes before using them in FAL/FCU.

Installing the FAL/FCU Program for UNIX

Caution *Before installing FAL/FCU on UNIX or Windows NT, make sure to deinstall any previous installations. See “FAL/FCU Deinstallation for UNIX” (page 77) and “FAL/FCU Deinstallation for Windows NT” (page 80) for instructions.*

Use the following procedures to install the FAL/FCU for UNIX by following the installation procedures below.

UNIX Installation for XP256 Only

It is an effective process only when a medium is FD/DAT.

1. Log-in as **root**.
2. Set the FD or DAT containing FAL/FCU to the drive.
(*Solaris only*) When using FD, create and check the device file for FD using the following commands.

volcheck

ls -l /vol/dev/diskette0

3. Determine if any of the devices listed in “FD/DAT Device Files” (page 75) exist.
4. Make sure the following directories exist; if not, create them.

/usr

/usr/bin

/usr/include

/usr/lib

/usr/lib/X11

/usr/lib/X11/app-defaults

<i>Example</i>	Create directory X11	# mkdir/usr/lib/X11
	Create directory app-defaults	# mkdir/usr/lib/X11/app-defaults
	5. Move to the root directory.	

6. Copy FAL/FCU from FD/DAT.
7. For the following platforms, use the additional instructions on the following page before continuing to the next step:
 UNIX systems (other than NCR UNIX and DYNIX/ptx)
 NCR UNIX
 DYNIX/ptx
8. Remove the FD/DAT from the drive.
9. *(Solaris only)* Enter the following command.
eject
10. Log out.
11. Log in again.

UNIX systems (other than NCR UNIX and DYNIX/ptx)

1. Copy the file:
cpio -iBmuv < device file name of FD/DAT (see “” (page 74))
Tru64 UNIX: uncompress FAL/FCU program
uncompress /usr/bin/fcu.Z
2. *(Solaris only)* To use FAL/FCU, you must use the path setting to the resource file. You do not need to set the path again after it is set.
 Use the following examples to set the path. When complete the setting below, log out and log in.

C shell only	All others
Add the following line to the last line of the file “.cshrc” in the home directory.	Add the following two lines to the last line of the file “.dtprofile” in the home directory.
If “.cshrc” does not exist, create and describe it.	If “.dtprofile” does not exist, create and describe it.
setenv XFILESEARCHPATH /usr/lib/X11/app-defaults/%N:\$XFILESEARCHPATH	XFILESEARCHPATH=/usr/lib/X11/app-defaults/%N:\$XFILESEARCHPATH
	export XFILESEARCHPATH

When not in the common desktop environment, add the two lines to the file “.profile” in the home directory, . If “.profile” does not exist, create it.

NCR UNIX Only

- Copy the file and uncompress FAL/FCU program.

cpio -icmuv < device file name of FD/DAT (see “FD/DAT Device Files” (page 75))

uncompress /usr/bin/fcu.Z

DYNIX/ptx Only

1. FTP the file.
2. Copy the file, then delete the .tmp file.
3. On the PC connected with NUMA-Q by LAN, in MS-DOS, enter:

C:\windows> a:

A:\> ftp <IP address of NUMA-Q>

ftp> cd /tmp

ftp> bin

ftp> put Seq.cpi

ftp> bye

4. At NUMA-Q, enter:

#cpio -iBmuv < /tmp/Seq.cpi

#rm /tmp/Seq.cpi

FD/DAT Device Files

The following shows the device files available for FD/DAT.

Table 1. Device files available for FD/DAT

Solaris	/vol/dev/diskette0/unlabeled	(FD)
	(underlined name is automatically determined)	
	/dev/rmt/0m	(DAT)
HP-UX	/dev/floppy/c0t1d0	(FD)
	/dev/rmt/0m	(DAT)
AIX	/dev/fd0	(FD)
DIGITAL UNIX/ Tru64 UNIX (before ver 4.0F)	/dev/fd0a	(FD)
	/dev/rmt0a	(DAT)
Tru64 UNIX (upper ver.5.0)	/dev/disk/floppy0c	(FD)
	/dev/tape/tape0c	(DAT)
NCR UNIX	:/dev/dsk/f03ht	(FD)
DYNIX-ptx	/dev/rmt/td0	(DAT)

UNIX Installation for XP512 Only

1. Log in as **root**.
2. Set the CD-ROM containing FAL/FCU to the drive.
3. Mount it.
4. Make sure the following directories exist; if not, create them.

/usr

/usr/bin

/usr/include

/usr/lib/usr/lib/X11

/usr/lib/X11/app-defaults

Example

Create directory X11

#mkdir/usr/lib/X11

Create directory app-defaults:

mkdir/usr/lib/X11/app-defaults

5. Move to the **root** directory.

6. Copy FAL/FCU from the CD-ROM.

The method of the mount determines how files/directories are seen.
(Capital letter/small letter and ‘;1’ is sometimes added to the file name.)

7. Install FAL/FCU after confirming a directory name and a file name by using the ls command.

(HP-UX only) #**cpio -iBmuv**

<(MountPoint)/PROGRAM/HMDE/HP_UX/HP_UX.CPI

(Solaris only) #**cpio -iBmuv**

<(MountPoint)/PROGRAM/HMDE/SOLARIS/SOLARIS.CPI

(AIX only) #**cpio -iBmuv**

<(MountPoint)/PROGRAM/HMDE/AIX/AIX.CPI

(Digital UNIX/Tru64 UNIX only) #**cpio -iBmuv**

<(MountPoint)/PROGRAM/HMDE/DIGITAL/"DIGITAL.CPI

(DYNIX only) #**cpio -iBmuv**

<(MountPoint)/PROGRAM/HMDE/DYNIX/DYNIX.CPI

(Solaris only) To use FAL/FCU, you must use the path setting to the resource file. You do not need to set the path again after it is set.

Use the following examples to set the path. When complete the setting below, log out and log in.

C shell only	All others
Add the following line to the last line of the file “.cshrc” in the home directory.	Add the following two lines to the last line of the file “.dtprofile” in the home directory.
If “.cshrc” does not exist, create and describe it.	If “.dtprofile” does not exist, create and describe it.
setenv XFILESEARCHPATH /usr/lib/X11/app-defaults/%N:\$X FILESEARCHPATH	XFILESEARCHPATH=/usr/lib/ X11/app-defaults/%N:\$XFILESE ARCHPATH

export XFILESEARCHPATH

When not in the common desktop environment, add the two lines to the file “.profile” in the home directory, . If “.profile” does not exist, create it.

8. Remove the CD-ROM from the drive.
9. Log out.
10. Log in again.

FAL/FCU Deinstallation for UNIX

If a previous version of FAL/FCU is already installed, use the following procedure to deinstall it.

Deinstalling FAL/FCU Installed by FD with an Installer

1. Log in with the user ID used at the installation.
2. Deinstall the FAL/FCU using the deinstaller, stored in FD of the older version FAL/FCU.

Deinstalling FAL/FCU Installed from DAT and with tar Command

Use the following procedure if you cannot locate the FD of the older version FAL/FCU (or if it is not available):

1. Log in with the user ID used at the installation.
2. Move to the **fcu/fal.o/dataset.h** directory, and deinstall it with the following command.

```
# rm fcu fal.0 dataset.h $HOME/FcuMf
```

If the directory of the older version is unknown, use the following procedure to locate it.

```
# find / -name "fcu" -print
```

```
# find / -name "fal.o" -print
```

```
# find / -name "dataset.h" -print
```

```
# find / -name "FcuMf" -print
```

Deinstalling FAL/FCU Installed with cpio Command

Use the following procedure if you need to deinstall FAL/FCU installed with **cpio** command.

Caution *You do not need to deinstall FAL/FCU for new FCU program installation. New installation overwrites the older program.*

1. Log in as **root**.

2. Execute the following commands:

```
# rm /usr/bin/fcu
```

```
# rm /usr/bin/fcunw
```

```
# rm /usr/include/dataset.h
```

```
# rm /usr/lib/libfal.*
```

```
# rm /usr/lib/libuoc.*
```

```
# rm /usr/lib/X11/app-defaults/FcuMf
```

```
# rm /usr/bin/mfformat
```

```
# rm /usr/bin/allocds
```

3. *(FAL/FCU Ver.41.XX.YY or later only)* Enter the following additional operation.

```
# rm /usr/bin/listvol
```

4. *(FAL/FCU for XP512 only)* Enter the following additional operation.

```
#rm /usr/bin/ppkeyset
```

Installing the FAL/FCU Program for Windows NT

If an older version of FAL/FCU program has already been installed, deinstall it first in accordance with the deinstallation procedure described later.

And then, install the new version of FAL/FCU by following the installation procedure below.

Windows NT Installation for XP256 Only

1. Set the FD containing FAL/FCU to the FD drive.
2. Run **setup.exe** in the provided FD.
The installation starts.
3. Follow the operation guidance on the screen to complete the installation.

Windows NT Installation for XP512 Only

1. Set the CD-ROM containing FAL/FCU to the CD-ROM drive.
2. Run \PROGRAM\HMDE\WIN_NT\DISK1\setup.exe from the CD-ROM.

The installation starts.

3. Follow the operation guidance on the screen to complete the installation.

FAL/FCU Deinstallation for Windows NT

1. Click Start | Settings | Control Panel.
2. Click Add/Delete Programs.
3. In the Install/Delete list, select FCU.
4. Click Add and Delete.

Deinstallation starts.

5. Follow the operation guidance on the screen to complete the deinstallation.

Inputting License Keys

You must use a license key for FAL for XP512.

Caution *License keys are not required for FAL for XP256.*

Error Messages

When FAL detects an error, it returns one of the following messages.

Table 2. Error messages from FCU/FCUNW/LISTVOL

	Error message	Action
1	Couldn't permit execution that does not have a software license.	Specify a proper license key code using ppkeyset command.
2	Couldn't permit execution that had an expired trying time limit of a software license.	Acquire a license for using this software product.
3	Couldn't permit execution that this host and a host installed are not identical and/or hostname is changed.	Execute these utilities using the host on which they were installed.

Setting License Keys

You can set license keys for UNIX and for Windows NT installations.

Setting License Keys for UNIX

1. On the command line, input the following command:

(HP-UX only) **ppkeyset <License key> /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.sl**

(Solaris only) **ppkeyset <License key> /usr/bin/fcunw /usr/bin/fcu/usr/lib/libfal.so.1**

(AIX only) **ppkeyset <License key> /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.a**

(Digital UNIX/Tru64 UNIX/DYNIX/ptx only) **ppkeyset <License key> /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.so**

Setting License Keys for Windows NT

1. In the install directory, run **ppkeyset.exe**.

MS-DOS displays the following message:

Please enter <License key> [Installed Directory or File]... or <quit> for exit.

2. Enter **<license key> < Installed Directory>**.
3. If < Installed Directory> is a directory name with “space”, enter the following:

<license key> fal.dll falmt.dll

MS-DOS displays the following message:

Please enter <License key> [Installed Directory or File]... or <quit> for exit.

4. Enter **quit**.

Definition Files

This section covers:

- “Volume Definition Files” (page 83)
- “Multiple Volume Definition Files” (page 93)
- “Parameter Definition Files” (page 95)

Volume Definition Files

You must use the volume definition file **datasetmount.dat** to relate logical volumes defined by the mainframe to volumes (i.e. partitions) defined by the open system.

Caution *Place the file in the directory from which you plan to run FAL/FCU.*

Specifying the Same VSN in Volume Definition Files

For v. 48.XX.20 or above only

You can use VSN identification to define the same VSN in the volume definition file.

Caution *Define the VSN identification with 35-digit alphanumeric (A-Z, 0-9, @, #, characters.*

Both Data Exchange OTM and Data Exchange MTO can use the volumes. Table 3 shows the relationship between volume definition files and parameter definition files.

Table 3. Relations between Volume Definition File and Parameter Definition File

Volume Definition File		Unspecified VSN identification	Specified VSN identification
Parameter definition file			
1	Not specified VSN Identification	The first volume is valid for the process.	The first volume is valid for the process.

Table 3. Relations between Volume Definition File and Parameter Definition File

Volume Definition File		Unspecified VSN identification	Specified VSN identification
Parameter definition file			
2	Specified VSN identification	None.	The first line to which VSN and VSN identification correspond

For v. 47.XX.00 or before only

You can define multiple volume information. For volumes containing duplicate VSNs, only the first volume is valid for the process.

SUN/ Solaris Volume File Definitions

Example

/dev/rdisk/cx1tx2dx3sx4	AAAAAA	3390-3A	MFN	MVS1	
/dev/rdisk/cy1ty2dy3sy4	AAAAAA	3390-3A	MFN	VOS3	
/dev/rdisk/cz1tz2dz3sz4	cccccc	3380-KB			
/dev/rdisk/cw1tw2dw3sw4	dddddd	3380-KA			
(1)	(2)	(3)	(4)	(5)	(6)
end					
(7)					

- 1 Defines a partition name defined in the open system.

The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.

Defines the multiple volume information can be defined. When volumes whose VSNs are duplicated, only the first volume is valid for the process.
- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.

- 3 Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported.
- 4 (*Reserve*) It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary.
- 5 (*v. 48.XX.20 or above only*) It is possible to omit this parameter. Specifies a VSN identification. It is an optional character line less than 36 characters.
- 6 Indicates each line above must be separated by pressing the Return key.
- 7 Specifies that the volume definition file ends here.
 - Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

HP-UX Volume File Definitions

Example

/dev/rdisk/cx1tx2dx3	AAAAAA	3390-3A	MFN	MVS	
/dev/rdisk/cy1ty2dy3	AAAAAA	3390-3A	MFN	VOS3	
/dev/rdisk/cz1tz2dz3	cccccc	3380-KB			
/dev/rdisk/cw1tw2dw3	dddddd	3380-KA			
	(1)	(2)	(3)	(4)	(5) (6)
end					
(7)					

- 1 Defines a partition name defined in the open system.

The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.

You can define multiple volume information. For volumes with duplicate VSNs, only the first volume is valid for the process.

- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.
- 3 Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported.
- 4 (*Reserve*) It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary.
- 5 (*v. 48.XX.20 or above only*) It is possible to omit this parameter. Specifies a VSN identification. It is an optional character line less than 36 characters.
- 6 Indicates each line above must be separated by pressing the Return key.
- 7 Specifies that the volume definition file ends here.
 - Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

AIX Volume File Definitions

Example

/dev/rhdiskn1	AAAAAA	3390-3A	MFN	MVS	
/dev/rhdiskn2	AAAAAA	3390-3A	MFN	MVS	
/dev/rhdiskn3	CCCCCC	3380-KB			
/dev/rhdiskn4	DDDDDD	3380-KA			
(1)	(2)	(3)	(4)	(5)	(6)
end					
(7)					

- 1 Defines a partition name defined in the open system.

The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.

You can define multiple volume information. For volumes with duplicate VSNs, only the first volume is valid for the process.
- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.
- 3 Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported.
- 4 (*Reserve*) It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary.
- 5 (*v. 48.XX.20 or above only*) It is possible to omit this parameter. Specifies a VSN identification. It is an optional character line less than 36 characters.
- 6 Indicates each line above must be separated by pressing the Return key.

- 7 Specifies that the volume definition file ends here.
- Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

Windows NT Volume File Definitions

Example

\\.\PHYSICALDRIVE0	AAAAAA	3390-3A	MFN	MVS	
\\.\PHYSICALDRIVE1	AAAAAA	3390-3A	MFN	VOS3	
\\.\PHYSICALDRIVE2	CCCCCC	3380-KB			
\\.\PHYSICALDRIVE3	DDDDDD	3380-KA			
(1)	(2)	(3)	(4)	(5)	(6)
end					
(7)					

- 1 Defines a partition name defined in the open system.
- The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.
- You can define multiple volume information. For volumes with duplicate VSNs, only the first volume is valid for the process.
- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.
- 3 Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported.

- 4 *(Reserve)* It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary.
- 5 *(v. 48.XX.20 or above only)* It is possible to omit this parameter.Specifies a VSN identification. It is an optional character line less than 36 characters.
- 6 Indicates each line above must be separated by pressing the Return key.
- 7 Specifies that the volume definition file ends here.
- Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

Digital UNIX and Tru64 UNIX Volume File Definitions

Example

/dev/rrzX1Y1c	AAAAAA	3390-3A	MFN	MVS	
/dev/rrzX2Y2c	AAAAAA	3390-3A	MFN	VOS3	
/dev/rrzX3Y3c	cccccc	3380-KB			
/dev/rrzX4Y4c	dddddd	3380-KA			
(1)	(2)	(3)	(4)	(5)	(6)
end					
(7)					

- 1 Defines a partition name defined in the open system.
- (Tru64 UNIX V4.0F or below only)*

Example

XnYm(n,m=1-6): Acquire it using file command.
(# file /dev/rrz*)

(Tru64 UNIX V5.0 only) The indication form is:

/dev/rdisk/dskX1cX1: Acquire it using file command.
(#file /dev/rdisk/dsk*)

The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.

You can define multiple volume information. For volumes with duplicate VSNs, only the first volume is valid for the process.

- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.
- 3 Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported.
- 4 *(Reserve)* It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary.
- 5 *(v. 48.XX.20 or above only)* It is possible to omit this parameter. Specifies a VSN identification. It is an optional character line less than 36 characters.
- 6 Indicates each line above must be separated by pressing the Return key.
- 7 Specifies that the volume definition file ends here.
 - Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

NCR UNIX Volume File Definitions

Example

/dev/rdisk/cx1tx2dx3s0	AAAAAA	3390-3A	MFN	MVS	
/dev/rdisk/cy1ty2dy3s0	AAAAAA	3390-3A	MFN	VOS3	
/dev/rdisk/cz1tz2dz3s0	CCCCCC	3380-KB			
/dev/rdisk/cw1tw2dw3s0	DDDDDD	3380-KA			
(1)	(2)	(3)	(4)	(5)	(6)
end					
(7)					

- 1 Defines a partition name defined in the open system.

The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.

You can define multiple volume information. For volumes with duplicate VSNs, only the first volume is valid for the process.
- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.
- 3 Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported.
- 4 (*Reserve*) It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary.
- 5 (*v. 48.XX.20 or above only*) It is possible to omit this parameter. Specifies a VSN identification. It is an optional character line less than 36 characters.
- 6 Indicates each line above must be separated by pressing the Return key.

- 7 Specifies that the volume definition file ends here.
- Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

DYNIX/ptx Volume File Definitions

Example

/dev/rdisk/sdx1	AAAAAA	3390-3A	MFN	MVS	
/dev/rdisk/sdx2	AAAAAA	3390-3A	MFN	VOS3	
/dev/rdisk/sdx3	cccccc	3380-KB			
/dev/rdisk/sdx4	dddddd	3380-KA			
(1)	(2)	(3)	(4)	(5)	(6)
end					
(7)					

- 1 Defines a device name defined in the open system.
- x1/x2/x3/x4 are device numbers recognized by the OS and put in this order.
- The FCU (actually the FAL) accesses a dataset of the mainframe by using raw IO. The file must be defined as a raw device.
- You can define multiple volume information. For volumes with duplicate VSNs, only the first volume is valid for the process.
- 2 Specifies a volume serial number defined in the mainframe with six-digit alphanumeric (A-Z, 0-9, @, #, and \) characters.

- | | |
|---|---|
| 3 | Specifies a device emulation type of the Data Exchange volume. See “Mainframe Device Emulation Types” (page 19) for device emulation types supported. |
| 4 | <i>(Reserve)</i> It is possible to omit this parameter. Specifies ‘MFN’ when the VSN identification is necessary. |
| 5 | <i>(v. 48.XX.20 or above only)</i> It is possible to omit this parameter. Specifies a VSN identification. It is an optional character line less than 36 characters. |
| 6 | Indicates each line above must be separated by pressing the Return key. |
| 7 | Specifies that the volume definition file ends here. |
- Each parameter must be separated with one or more “space” character(s).
 - The underlined portion is a variable field corresponding to the actual environment.
 - Lines containing “#” in the head position are assumed to be comment lines.

Multiple Volume Definition Files

Caution *Only Data Exchange MTO V50.XX.YY(XX=01 or 02) or above supports multiple volume datasets.*

The multiple volume definition file name is **multidef.dat**. Place the file in the directory from which you plan to run FAL/FCU.

To change the name of the multiple volume definition file, specify the name in FAL_MULTI_DEF_FILE of the environment variable.

Example

VSN:DSN [, VOLID1]	VSN [, VOLID2]	----	VSN [, VOLIDn]
1)	2)	3)	4)
end			
5)			

- 1 Contains the head volume's information.

 VSN: Volume serial number with six alphanumeric (A-Z, 0-9, @, #, and \) characters.

 DSN: Dataset name. Use a maximum of 44 alphanumeric characters.

 VOLID: VSN identification. Specify the same as the volume definition file, if a VSN identification is specified in the volume definition file, or omit if not specified.
- 2 Contains the second volume's information.

 VSN: Volume serial number with six alphanumeric (A-Z, 0-9, @, #, and \) characters.

 VOLID2: VSN identification Specify the same as the volume definition file, if a VSN identification is specified in the volume definition file, or omit if not specified.
- 3 Contains the last volume's information. (The number of volume is 'n')

 VSN: Volume serial number with six alphanumeric (A-Z, 0-9, @, #, and \) characters.

 VOLIDn: VSN identification Specify the same as the volume definition file, if a VSN identification is

	specified in the volume definition file, or omit if not specified.
4	Indicates each line above must be separated by pressing the Return key.
5	Specifies that the volume definition file ends here.
<i>Notes</i>	<ul style="list-style-type: none"> • Each parameter must be separated with one or more “space” character(s). • One data set information must be specified in one line. • 999 information can be specified in the multiple volume definition file.

Parameter Definition Files

The initiation parameters for the FCU are defined using the following parameter format.

The default file names are:

- (UNIX) **fcudata.param**
- (Windows NT) **fcudata.prc** with “-nc” setting

You can either create this file interactively by using the FCU, or in advance using an editor.

Caution *You do not need to use the parameter definition file if the parameters are always set by operation screen.*

UNIX Parameter File Definitions

You can define a maximum of 999 parameter sets within one parameter definition file.

Caution *If you define more than 999 parameter sets, UNIX rejects a parameter set number of 1,000.*

Lines with “#” in the head position are assumed to be comment lines.

GUI Environment Only	All Others
When a comment line is loaded, contents of the comment line are not displayed and “Parameter file: Comment line” is displayed in Status.	A comment line is skipped. A comment line is not counted for “maximum 999 parameter sets”.
A comment line cannot be made by GUI operations.	
You can delete and change a comment line to a normal parameter line by GUI operations.	
A comment line is counted for “maximum 999 parameter sets”.	

Windows NT Parameter File Definitions

You can define a maximum of 999 parameter sets within one parameter definition file.

Caution	<i>The memory capacity of the server determines which number of parameter sets exceeding 999 Windows NT rejects.</i>
----------------	--

Comment Lines

- Comment lines are skipped and cannot be processed.
- Comment lines are not included in the line count.
- Comment lines cannot be created, deleted, or changed to valid lines through the FCU GUI screen.
- Use an editor to modify comment lines.

Data Transfer Process

The defined parameter set determines the process of each data transfer.

Example

mtm	VSN:dataset name	Open system file name	CC	PAD	DEL	Emp=Yes	RDW=Yes	PIPE=No	VSE=RF,RL,BL	
mtm	VSN:dataset name	Open system file name	CC	PAD	DEL	Emp=Yes	RDW=Yes	PIPE=Yes	VSE=RF,RL,BL	
otm	Open system file name	VSN:dataset name	CC	PAD	DEL	Emp=Yes			VSE=RF,RL,BL	
otm	Open system file name	VSN:dataset name	CC	PAD	DEL	Emp=Yes			VSE=RF,RL,BL	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
end										
(12)										

1 Specify the one of the following data transfer directions:

MTO: mainframe to open system

OTM: open system to mainframe

2 Specify the input mainframe dataset name or open system file name.

Mainframe Dataset File Names	Open System File Names
<VSN (Volume Serial Number)>: < dataset name>(VSN identification>)	Specify the name using one of the following formats. "Space" characters are not valid.
VSN: Use six alphanumeric characters.	(UNIX only) /directory name/...../directory name/file name (with path) file name (without path)
Dataset name: Use a maximum of 44 alphanumeric characters. "Space" characters are not valid.	
VSN identification: Use a maximum of 35 alphanumeric characters.	(Windows NT only) drive:\\directory name\\.....\\directory name\\file name (with path) file name (without path)
It is possible to omit this parameter.	
	If you only specify the file name, the file in the current directory is selected.

3 Specify an output mainframe dataset name or an open system file name.

Use the same format as defined for the input dataset name or file name.

4 **CC**: Specifies the code conversion type of the data entity. An error is reported when the defined file does not exist.

Specify **No** when using Data Exchange OTO. Code converting function is not supported for Data Exchange OTO; even if you specify **EA** or the file name of the conversion table in this field when using Data Exchange OTO, the code does not convert.

(Code conversion table built in the FCU only) Specify one of the following:

- EA or EcA: EBICDIC in mainframe to/from ASCII in open system.

- No: No code conversion is applied.

(User-defined code conversion table only) Specify the file name of the conversion table, using the following guidelines:

The code conversion table must be 256 bytes in size and must be binary data.

Only one byte conversion is supported. Two bytes code cannot be converted.

The code conversion table is created as putting a target code into the offset position in the table which represents the code value of the source code to be converted.

Do not use “EA”, “EcA”, “EkJ”, or “No” as part of or as the full file name. The conversion table built in the FCU is automatically used if any of these characters are used.

5

PAD: Specify one of the following:

Yes: Padding is applied.

No: No padding is applied.

6 DEL: Specifies one or more of the following delimiters.

	UNIX	Windows NT
“MTO” data transfer direction	<p>CR: Carriage return is added as a delimiter.</p> <p>LF: Line feed is added as a delimiter.</p> <p>No: No delimiter is added.</p>	<p>CRLF: CR+LF is added as a delimiter.</p> <p>No: No delimiter is added.</p>
“OTM” data transfer direction	<p>CR: Data up to carriage return is cut off as a data entity.</p> <p>LF: Data up to line feed is cut off as a data entity.</p> <p>No: Data is cut off according to the dataset record length.</p>	<p>CRLF: CR+LF is added as a delimiter.</p> <p>No: No delimiter is added.</p>

The data edit format depends on whether or not PAD is specified if the dataset’s record format is a fixed length and a value other than **No** is specified. Refer to section “Padding Functions” (page 60) for further detailed specification.

7 Specify whether or not an one of the following empty files is transferred:

Mainframe dataset: No data records or EOF records are written in the dataset

Open system file: File size=0

This parameter can be omitted. If not specified, it is assumed to be Emp=No.

Emp=Yes: The data transfer is executed even if the source file is an empty file. When it is transferred to open system, a file with file size=0 is created. When it is transferred to mainframe, only a dataset with EOF is created.

Emp=No: The data transfer is not executed and rejected with an error if the source file is an empty file.

- 8 Specify if a record length is added to create an open system file. This is only valid when the transfer direction is mainframe to open and the mainframe dataset format is a variable record length.

This parameter can be omitted. If not specified, it is assumed to be RDW=No.

RDW=Yes: Record length is added to the head of each record in binary code. In this case, specify **No** for the CC, PAD, and DEL parameters, or an error is returned.

An error is returned when “OTM” is combined with RDW=Yes.

When fixed length record format is specified, RDW=Yes is ignored and RDW is not added.

Record Length	Dummy	Record (Data) entity
2 bytes	2 bytes	Record length - 4 bytes
Record length		

RDW=No: Record length is not added. Only the data entity will be output.

Note When the data is restored to mainframe after transferred from mainframe to open system with setting of RDW=Yes, the dataset is created but is incompatible with the original dataset.

- 9 Specify transfer using a named pipe or not when the data transfer direction is MTO.

Pipe is for UNIX use only

It is possible to omit this parameter. When it is not specified, it is assumed to be PIPE=No

PIPE=Yes: Transfer using a named pipe

An error is reported when the data transfer direction is OTM

PIPE=No: Transfer to a file

10 Specify record format, record length, and block length for a dataset that does not define these parameters. An error is reported if other than the following is specified.

VSE: Specify record format, record length, and block length. When these parameters are not specified, suppress the “VSE=RF,RL,BL” field. An error is reported if other than the following is specified.

RF: Specifies the record format of the dataset.

F: Specifies fixed length and un-blocking

FB: Specifies fixed length and blocking

V: Specifies variable length and un-blocking

VB: Specifies variable length and blocking

RL: Specifies record length of dataset in decimal and in bytes.

When RF = F, record length = block length

When RF = FB, record length = (block length / n). n: integer

When RF = V or VB, $5 \leq \text{record length} \leq (\text{BL} - 4)$

BL: Specifies block length of dataset in decimal and in bytes.

When RF = F or FB, 1 through 32760 is allowed

When RF = V or VB, 9 through 32760 is allowed

Each parameter must be separated by one comma “,”.

When the record format, record length, and block length are not specified in the VTOC, FAL/FCU program considers it a VSE dataset. An error is reported if the parameters (i.e. RF, RL and BL) are not set for the VSE dataset.

When the record format, record length, and block length are specified in the VTOC, specified parameters are checked if they are the same. If they are the same, FAL/FCU starts the process; otherwise, an error is reported.

Do not use this parameter with Data Exchange OTO. (FCU moves in the same way as the Data Exchange MTO or Data Exchange OTM, if it is specified.)

11

Separate each line by:

(UNIX) Press the Return key

(Windows NT) Press the Enter key

- 12 end: Specifies the end of parameters. This parameter can be omitted.
- Each parameter must be separated by one or more “space” character(s).
 - No space line is allowed from the beginning to the end of the parameter lines. If “-nw” is specified within the initiation parameter, space lines are skipped and processed without error.
 - As defined above, parameters other than “Emp”, “RDW”, and “VSE” must be specified in order. You can freely order “Emp”, “RDW”, and “VSE”, but do not place them before other parameters.
 - You cannot omit parameters (1) to (6). An error is reported if fewer than six parameters exist. Additionally, if more than 11 parameters are specified, parameters after the 11th are ignored.

Starting the File Conversion Utility

After the environment setting, such as FAL installation or volume definition, is complete, you can start the FCU by inputting the following command from the command line.

You can also create the parameter definition file after starting the FCU.

Starting FCU for UNIX

Starting FCU for UNIX Using GUI Environment

- To start the FCU, input the following command from the command line.

fcu [-nw] [-nc] [param]

<i>Command Details</i>	fcu	Indicates the FCU program name. Input as fcu .
	-nw	Used when the utility’s operation screen is not used. When this parameter is specified and other parameters are not directly specified in the param (see description further in this detail), all the lines (i.e. maximum 999 lines) in the parameter definition file are processed continuously. Input as -nw . When this parameter is not specified, the operation screen (figure 2 (page 141) and figure 3 (page 144)) opens. You can specify, check, and modify the parameter on the screen, to execute file conversion and check the execution result.
	- nc	Used when the parameter check is not needed. When this parameter is specified, the FCU program immediately start the process for all parameter lines continuously without checking parameters. Input as -nc .

When this parameter is not specified, initiate the conversion process by inputting **OK** in the parameter display.

param Specifies a file name, **parameter definition file**, in which the initiation parameters for the FCU are stored. You can also directly specify the detail parameters to be executed.

When this parameter is not specified, **parameter definition file** is assumed to be **fcudata.param** and it attempts to read the detail parameter from the file. The FCU terminates operation when **-nw** or **-nc** is specified and **parameter definition file** does not exist.

- You can use either an absolute or relative path when using the file name **parameter definition file**.

Example Absolute path: **/directory name/directory name/file name**

Relative path: **file name ./file name**

directory name/file name ./ directory name/file name

- When directly specifying the detail parameters, set **-P**, then follow with the detail parameters as same format as defined in the **parameter definition file**.

Example **# fcu -nw -nc -P mto VSNAAA:DSNAME outfile No No No**

- This direct setting function is not available for Windows NT environment.
- Parameter must be separated by one or more “space” character(s).
- Only the detail parameters can be set after **-P**. **-nw** or **-nc** must precede **-P**.
- Always set **-nc** when FCU starts **-nw** in the background process. Otherwise sometimes FCU stops and waits for key input.

When the operation terminates, the following return value is reported.

[Return value]

0 : Normal end
 -1 : Error end

Starting FCU for UNIX When Not Using GUI Environment

This has the same function as specified **-nw** in the GUI environment described in “Starting FCU for UNIX Using GUI Environment” (page 105).

- To start the FCU, input the following command from the command line.

fcunw [-nc] [param]

<i>Command Details</i>	fcunw	Indicates the FCU program name when executing it without GUI. Input as fcunw .
	- nc	Used the same as in “Starting FCU for UNIX Using GUI Environment” (page 105).
	-param	Used the same as in “Starting FCU for UNIX Using GUI Environment” (page 105).

Always set **-nc** when **fcunw** is started in the background process.

Starting FCU for Windows NT

1. Log in using a user ID with administrator privileges.
2. *(Windows only)* To initiate the program, click Start | Programs | FCU, then click the FCU icon.

You can also move to C:\FCU, the default directory, and start FCU, or make a shortcut to FCU and double-click the icon.

3. *(DOS prompt only)* Open the command prompt and input the following commands and parameters.

fcu [-nc] [-cl] [param]

<i>Command Details</i>	<code>-nc</code>	Used the same as in “Starting FCU for UNIX Using GUI Environment” (page 105).
	<code>-cl</code>	Specifies that the FCU log file will be cleared before stating FCU.
	<code>param</code>	Used the same as in “Starting FCU for UNIX Using GUI Environment” (page 105). When this parameter is not specified, the parameter definition file is assumed to be fcudata.prm , and it attempts to read the detail parameters from the file.

Return Value

0 : Normal end
-1 : Error end

- The return value is only valid when “-nc” is specified.
- Even if some of the command lines complete successfully, any single error returns the error in the return value.
- Do not drag and drop the parameter definition file onto the FCU icon. The operation and function cannot be guaranteed.

FCU OPERATIONS AND SCREENS FOR WINDOWS NT

This chapter describes the interface and operation instructions for using File Conversion Utility (FCU) with Windows NT.

FCU Interface

This section describes the elements of the FCU screens for Windows NT, including menus, fields, buttons, and check boxes.

Splash Screen

When you start the FCU, the splash screen opens for a few seconds, then closes.

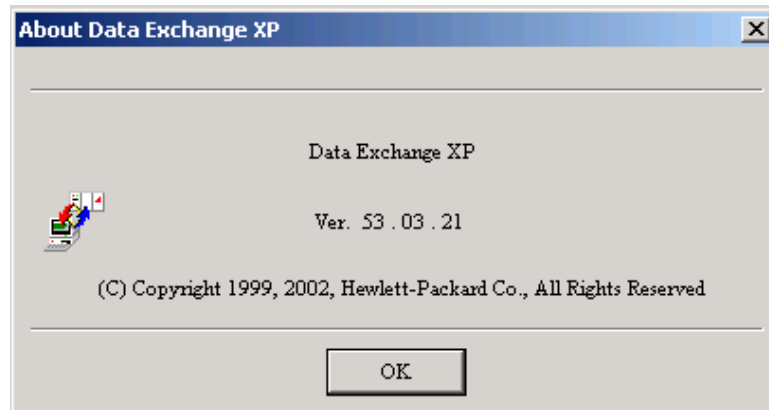


Figure 1. Splash screen

Main Screen

After the splash screen closes, the Main screen opens.

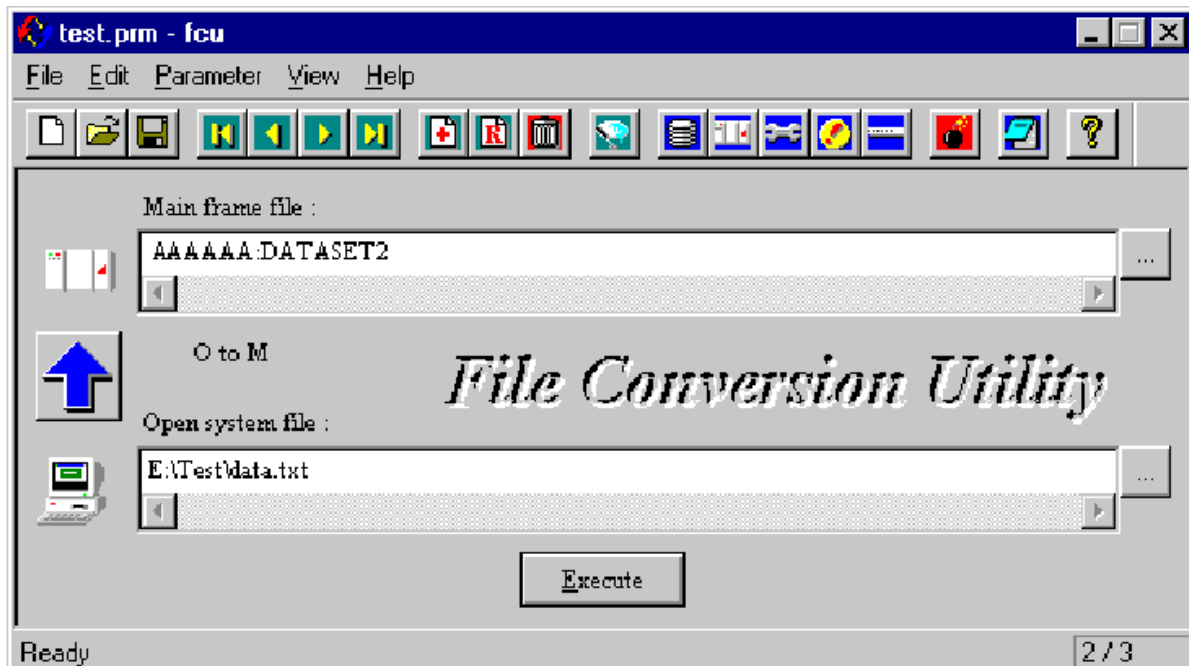


Figure 2. FCU main dialog box

The Main screen contains the following menus:

File	Provides access to operations for executing and stopping the parameter definition file and FCU operation.
Edit	Reserved for the future enhancement. No submenus appear.
Parameter	Operates the current parameter definition file, or clears the parameters on the screen.
View	Allows you to view or hide the tool bar and status bar; change options; close all dialog boxes; and display mainframe file, error, volume, and log file information.
Help	Displays version information.

File Menu

This menu provides access to create, open, save, and close parameter definition files.

This menu contains the following submenus:

New	(Ctrl + N) Creates a new parameter definition file. The parameter definition file contents are stored in memory and not reflected on a disk drive until you click Save.
Open	(Ctrl + O) Opens an existing parameter definition file.
Save	(Ctrl + S) Saves the parameter definition file to a disk drive. You can only save the contents of the Parameter line dialog box. Any parameter lines located after the end line are discarded.
Save As	Saves the parameter definition file with another file name.
Exit	Terminates the FCU process.

Edit Menu

This menu is reserved for the future enhancement.

Parameter Menu

Select this menu to operate the parameter definition file being edited on memory or clear the parameters on the screen.

This menu contains the following submenus:

Load	<p>Performs the following operations:</p> <p>Previous: Loads the previous parameter line.</p> <p>Next: Loads the next parameter line.</p> <p>Top: Loads the top parameter line.</p> <p>Bottom: Loads the bottom parameter line.</p> <p>The parameter's position is the line number in the parameter definition file (also indicated at the bottom right in the status bar).</p>
Save	<p>Saves the parameter definition file with the following operations:</p> <p>Insert: Inserts the new parameter line, and saves it into the next parameter line to the current position.</p> <p>Replace: Replaces the current parameter, saving the new parameter in the current position.</p> <p>Use this option to save when you make changes to any registered parameters through the Main screen or Option dialog box.</p>
Delete	<p>Deletes the current parameter.</p>
<hr/> Caution <i>The parameter is not actually deleted until you click Save.</i> <hr/>	
Wipe	<p>Clears the displayed parameter contents.</p> <p>The current parameter is not deleted with this function.</p>

View Menu

Select this menu to view or hide the tool bar and status bar; change options; close all dialog boxes; and display mainframe file, error, volume, and log file information.

This menu contains the following submenus:

Tool bar Displays or hides the tool bar.

Status bar Displays or hides the status bar.

Volume Information Opens the Volume information dialog box.

MF-file Information Opens the Mainframe Information dialog box. See “MF-file Submenu” (page 114) for a detailed description.

Option Opens the Option dialog box. See “Option Submenu” (page 116) for a detailed description.

Error Information Opens the Error Information dialog box. See “Error Information Submenu” (page 120) for a detailed description.

MF-file Submenu

The MF-File select dialog box enables you to view mainframe information.

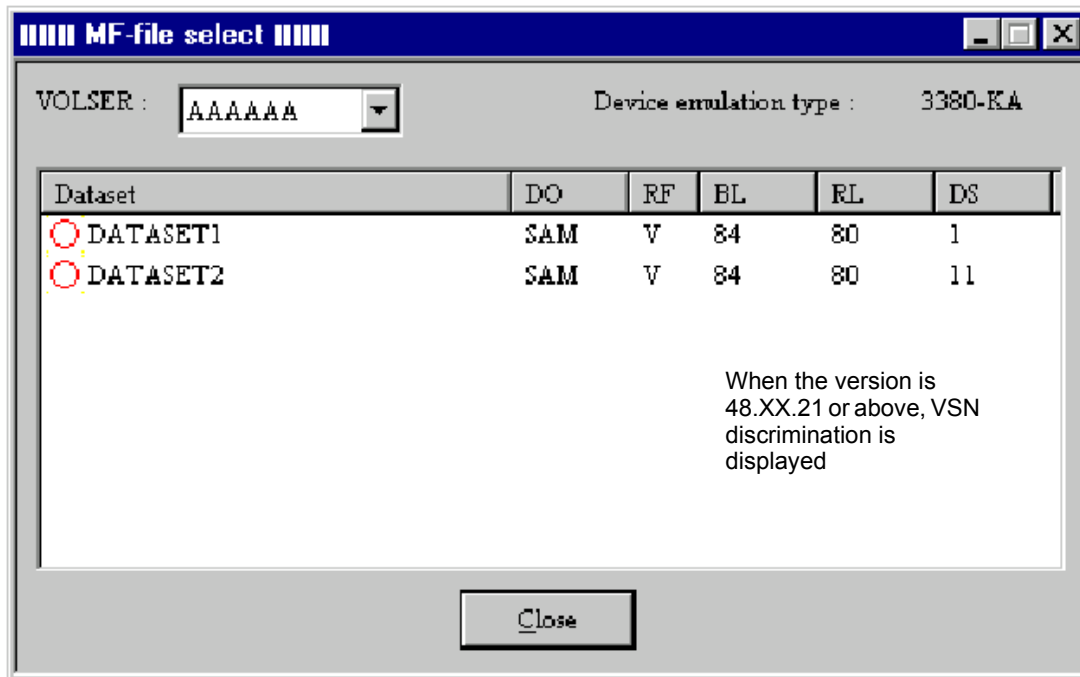


Figure 3. MF-File select dialog box

Dataset validation ☐ Indicates the dataset can be processed by FCU.

☒ This shows the dataset cannot be processed by FCU.

Dataset Displays the dataset name

DO Displays the dataset organization type:

SAM: Indicates sequential access method organization

DAM: Indicates direct access method organization

PAM: Indicates partitioned access method organization

VSAM: Indicates virtual storage access method organization

RF Displays the record format

F: Indicates fixed-length record

FB: Indicates fixed-length and blocking record

V: Indicates variable-length record

VB: Indicates variable-length and blocking record

U: Indicates undefined-length record

S: Indicates spanned record

?: Indicates a format not described above

BL Displays the block length

RL Displays the record length

DS Displays the dataset size in number of tracks

The total of all volumes displays, when the head
volume of multiple volumes is specified.

Option Submenu

The Option dialog box enables you to select or change options. You can select each field value manually or from the menu.

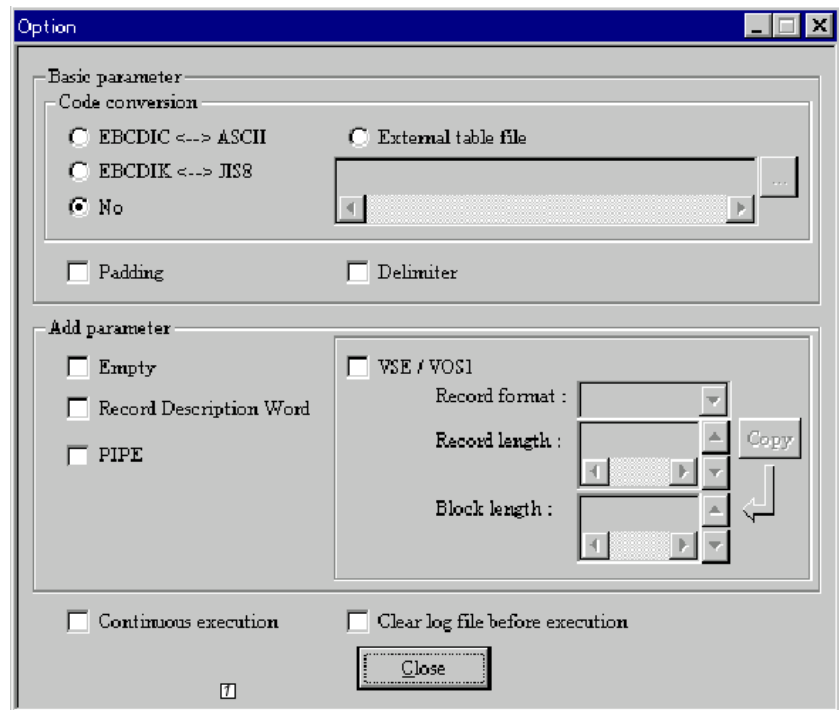


Figure 4. Option dialog box

Code conversion Indicates whether or not to apply the code conversion function.

EBCDIC <--> ASCII: Converts code between EBCDIC and ASCII formats.

EBCDIC < -- > JIS8: converts code between EBCDIC and JIS8 (Japanese).

No: No code conversion is applied.

External table file: Converts code using a user-specified conversion table. Select the radio button, then type the conversion table file name.

Padding	Indicates whether or not to apply padding function. See “Code Converting Functions” (page 58) for detailed specification.
Delimiter	Indicates whether or not to apply delimiter function. See “Padding Functions” (page 60) for detailed specification.
Empty	Indicates whether or not to transfer an empty file (i.e. file size equal to zero). See “Parameter Definition Files” (page 95) for detailed specification.
Record Description Word (RDW)	Indicates whether or not to add RDW to the head of each data and to create the open system file. See “Parameter Definition Files” (page 95) for detailed specification.
	The following settings are ignored when this check button is selected.
	EBCDIC <---> ASCII radio button
	Padding check box
	Delimiter check box
PIPE	Not available.
Continuous execution	Indicates whether or not to execute the operation continuously.
	Continuous operation runs from the current parameter line to the end of parameter line.
Clear log file before execution	Indicates whether or not to clear the log file before starting the operation.
VSE/VOS1	Processes the VSE dataset and specifies the dataset’s record format, record length, and block length.

Caution *Do not use this parameter when using Data Exchange OTO.*

Record length (RF): Specifies the dataset record format:

F: Indicates fixed length and un-blocking

FB: Indicates fixed length and blocking

V: Indicates variable length and un-blocking

VB: Indicates variable length and blocking

Record length (RL): Specifies the dataset record length in decimal and in bytes.

When RF = F, R= F, record length = block length.

When RF = FB, record length = (block length / n).
n: integer

When RF = V or VB, $1 \leq \text{record length} \leq \text{BL}$

An error is reported if you specify other than above.

Block length (BL): Specifies the dataset block length in decimal and in bytes.

When RF = F or FB, 1 through 32760 is allowed.

When RF = V or VB, 9 through 32760 is allowed.

An error is reported if you specify other than above.

Error Information Submenu

The Error Information dialog box enables you to view parameter line error information.

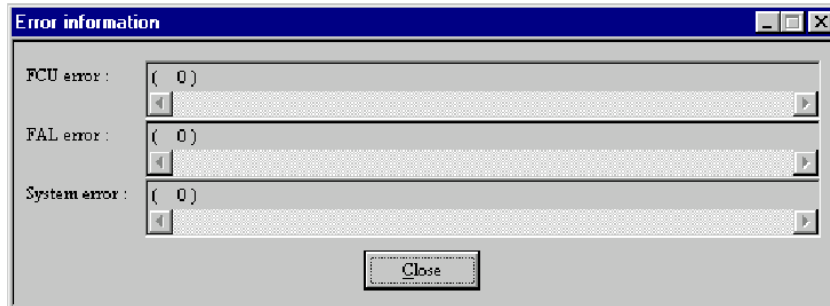


Figure 5. Error information dialog box

FCU Error Indicates an FCU error number and error message

FAL Error Indicates an FAL error number and error message

System Error Indicates a system error number and error message

Parameter Line Opens the Parameter line dialog box

Close All Dialog Closes all dialog boxes

Log File Opens the Log file dialog box

Logs are saved with a parameter definition file name and a .log extension, such as Test.prm.log

Caution *If log files are not cleared, the disk drive can fill. Select the Clear log file before execution check box to clear the log files.*

Help Menu

The Help menu displays version information.



Figure 6. About FCU dialog box

Tool Bar

The tool bar contains the following buttons:

Table 1. Tool bar buttons






Button	Name	Function
	New	same as clicking File New
	Open	same as clicking File Open
	Save	same as clicking File Save
	Top	same as clicking Parameter Load Top
	Previous	same as clicking Parameter Load Previous

Table 1. Tool bar buttons (continued)















Button	Name	Function
	Next	same as clicking Parameter Load Next
	Bottom	same as clicking Parameter Load Bottom
	Insert	same as clicking Parameter Save Insert
	Replace	same as clicking Parameter Save Replace
	Delete	same as clicking Parameter Delete
	Wipe screen	same as clicking Parameter Wipe
	Volume info	same as clicking View Volume information
	MF-file info	same as clicking View MF-file information
	Option	same as clicking View Option
	Error info	same as clicking View Error information
	Parameter line	same as clicking View Parameter line
	Close all dialogs	same as clicking View Close all dialogs

Table 1. Tool bar buttons (continued)

Button	Name	Function
	Log	same as clicking View Log file
	Help	same as clicking Help About FCU

Buttons and Fields

This section describes the buttons and fields in the Main screen.

Direction buttons



Transfers data from mainframe to open system (MTO).



Transfers data from open system to mainframe (OTM).

Mainframe file field

Displays the mainframe file to be transferred.

- Enter the file name using the following format:
volume serial number:dataset name

Example **AAAAAA:DATASET**

- The volume serial number is a fixed length six-digit number
- The dataset name is an 44-character or less alphanumeric combination

Open system file field

Displays the name of the open system file to be transferred.

Caution *Do not use spaces in an open system file name or a path name to the file, or an error is generated when the parameter line is read, because space characters are used as separators for each parameter in the parameter definition file.*

Mainframe file selection button

Click this button to open the Mainframe file selection dialog box.

Open system file selection button

Click this button to open the Open system file selection dialog box.

Execute button

Click this button to start the data transfer.

Status Bar

The status bar indicates the current parameter position and total number of parameters.

e.g.; A/B: A---The current parameter position.

B---Total number of parameters.



Figure 7. Status bar

FCU Operations

This section describes the file conversion utility (FCU) operations process for the Windows NT platform, including creating a sample parameter definition file (Test.prm), and converting the file using the parameters.

In addition, a detailed description of the Main screen is provided in “FCU Interface” (page 110).

The process is broken down into the following procedures:

- “Creating Volume Definition Files for FCU” (page 125)
- “Starting the FCU” (page 126)
- “Creating New Parameter Definition Files” (page 128)
- “Entering Parameter Line Data” (page 128)
- “Saving Parameter Definition Files to Disk” (page 136)
- “Opening Existing Parameter Definition Files” (page 137)
- “Executing Parameter Lines” (page 137)

Assumptions

- Datasets (DATASET1 and DATASET2) exist in the mainframe’s VOLSER (AAAAAA)
- An open system file (E:\test\data.txt) exists and conforms with the organization of DATASET2 and record format
- The physical connections for both the SCSI cable and logical connection are complete
- All equipment, such as the open system host and DKC, is powered on and working normally

Creating Volume Definition Files for FCU

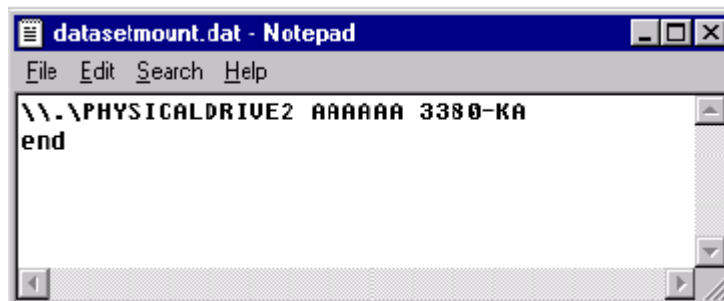
Before you can start the FCU, you need to make sure the XP256/XP512 volumes are recognized, and create the volume definition file.

To verify the XP256/XP512 volumes are recognized

1. Log in using a user ID with administrator privileges.
2. Click Start | Settings | Control Panel.
The Control Panel window opens.
3. Double-click SCSI Adapters.
4. Verify that the XP256/XP512 volumes are recognized.

To create the volume definition file

1. Click Start | Programs | Administrative Tools (Common) | Disk Administrator.
2. Check the physical disk number.
3. Create the volume definition file.

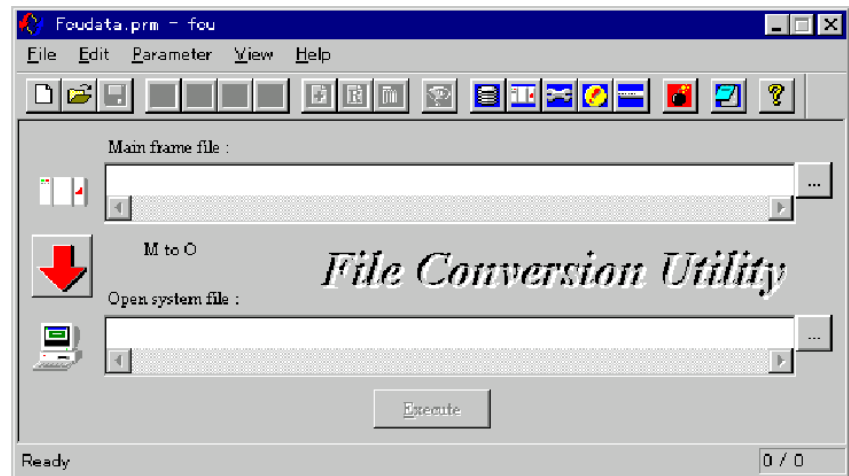



Starting the FCU

When you start the FCU, you need to verify the volume information in the Volume Information screen, and make changes as necessary, before entering the first parameter line.

To start the FCU

- Click Start | Programs | Fcu | fcu.
The FCU screen opens.

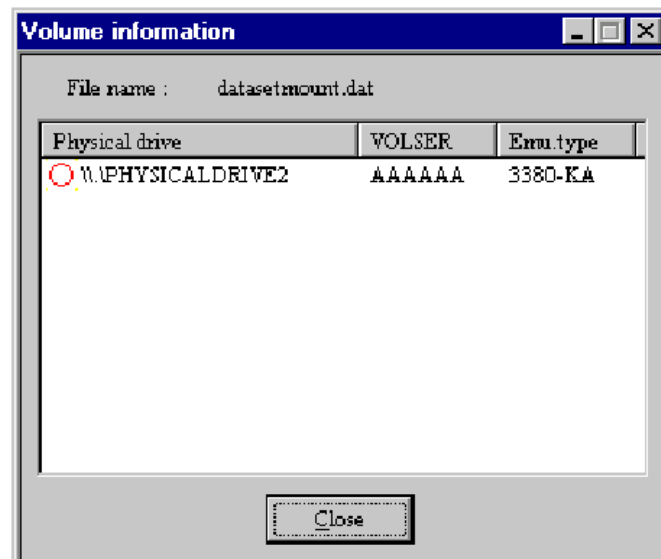


To close the FCU screen, click the  in the top right corner of the screen.

To verify volume definition information

1. Click the Volume Info  button.


The Volume information screen opens.



2. Make sure that “○” displays next to the path name.

If “✕” displays, modify the volume definition file.

Creating New Parameter Definition Files

To create a new parameter definition file, click  on the toolbar.

FCU automatically creates a new file when you start the program.

Entering Parameter Line Data

To set up the parameter lines for the sample FCU (described in “Assumptions” (page 125)), you need to create the lines sequentially, selecting the mainframe file, open system file, and parameter definitions for each.


Entering the First Parameter Line

To create the first parameter line, you need to select the mainframe and open system files, then identify the parameters.

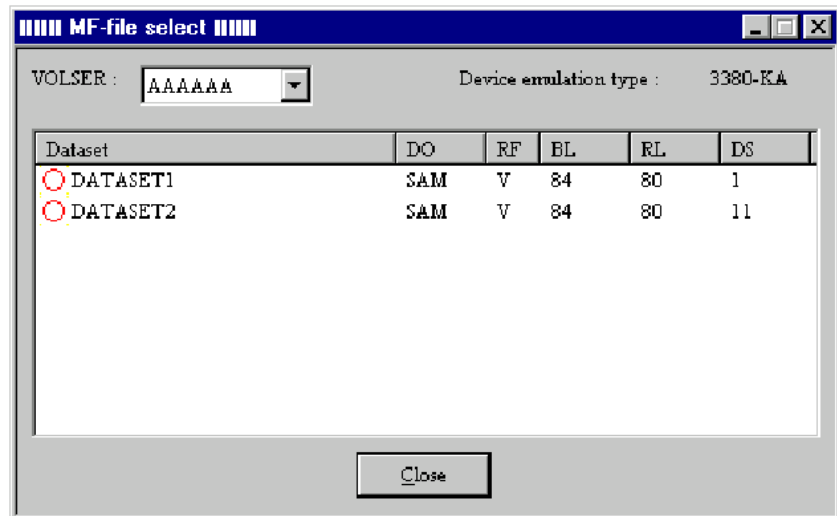
To select the mainframe file

1. In the FCU main screen, make sure the data transfer direction is set to MTO.



2. Next to the Main frame file field, click the select  button.

The MF-file select dialog box opens.




3. In the VOLSER list, select AAAAAA.

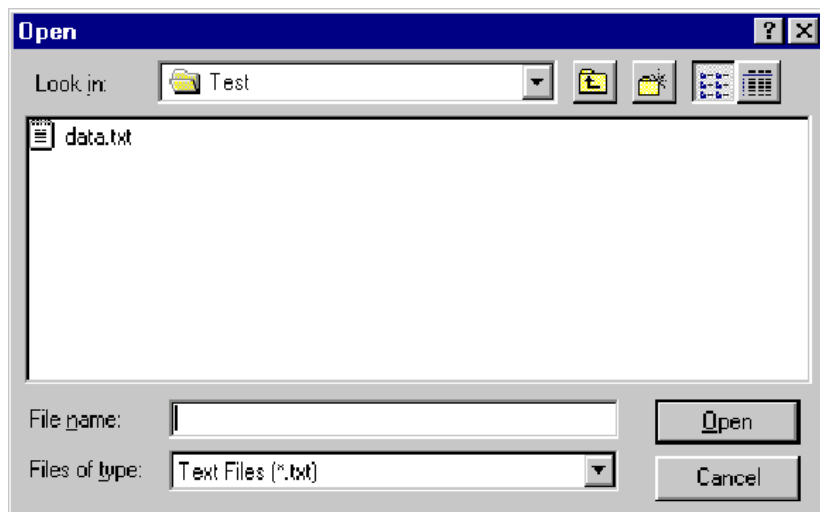
4. Double-click DATASET1.

The MF-file select dialog box closes, and **AAAAA: DATASET1** displays in the Main frame file field.

To select the open system file

1. Next to the Open system file field, click the select  button.

The Open dialog box opens.




Because the data transfer direction is MTO, no open system files exist, so the file name out1.txt does not display in the dialog box.

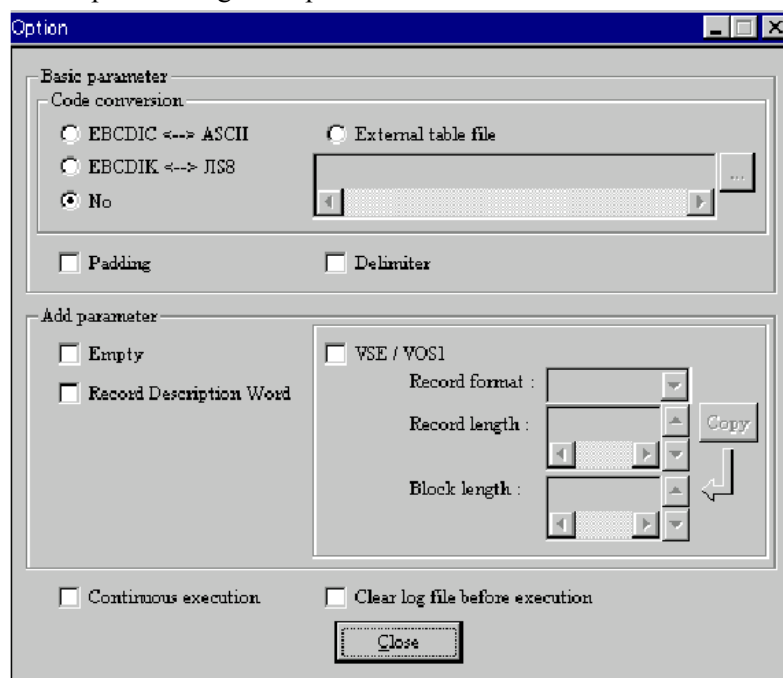
2. In the File name field, type **out1.txt**, then click Open.

The Open dialog box closes, and the path name including the file displays in the Open system file field.


To select the first line parameter options

1. In the FCU main screen, click the Option  button.

The Option dialog box opens.



2. Under Code conversion, select the EBCDIC <--> ASCII radio button.
3. Select the Delimiter check box.
4. Click Close to close the Option dialog box.

5. In the FCU main screen, click the Insert  button.

When you click Insert, the data you enter in the FCU main screen and in the Option dialog box are registered in the parameter definition file (which is still being edited on the memory).

The numbers on the right side of the status bar change from “0/0” to “1/1” to indicate that the parameter entry has been registered.

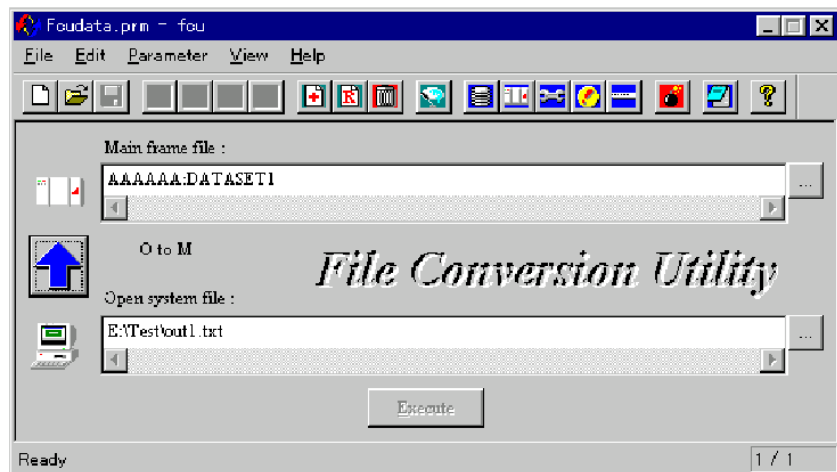
6. Click Save.


Entering the Second Parameter Line

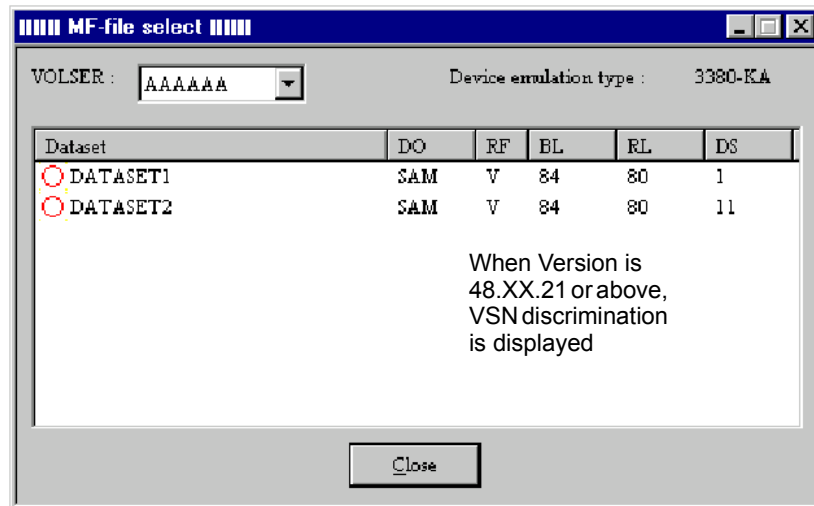
To create the second parameter line, you need to change the data transfer direction, select the mainframe and open system files, and then identify the parameters.

To select the data transfer direction and mainframe file

1. In the second parameter line, click the data transfer direction button to reverse the direction to OTM.




2. Next to the Main frame file field, click the Select  button.
The MF-file select dialog box opens.



3. In the VOLSER list, select AAAAAA.
4. Double-click DATASET2.

The MF-file select dialog box closes, and **AAAAA: DATASET2** displays in the Main frame file field.

To select the open system file

1. Next to the Open system file field, click the select  button.
The Open dialog box opens.



2. Double-click data.txt.

The Open dialog box closes, and the file name displays in the Open system file field.

Selecting the Second Parameter Line Options

You do not need to open the Options dialog box for the second parameter line, as the options are the same as for the first parameter line. For information, see “To select the first line parameter options” (page 130).

3. Click the Insert  button.

The parameter line is registered into memory. The numbers on the right side of the status bar change from “1/1” to “2/2” to indicate two parameter entries are registered.

4. Click Save.

The parameter lines are not saved until you click Save.

Entering the Third Parameter Line

To create the third parameter line, you need to change the data transfer direction, select the mainframe and open system files, and then identify the parameters.

To select the data transfer direction and mainframe file


1. In the third parameter line, click the data transfer direction button to reverse the direction to MTO.



Selecting the Mainframe File

You do not need to change the mainframe file for the third parameter line, as it is the same as for the second parameter line.

To select the open system file


1. Next to the Open system file field, click the select  button.
2. The Open dialog box opens.



Because the data transfer direction is MTO, no open system files exist, so the file name out2.txt does not display in the dialog box.

3. In the File name field, type **out2.txt**, then click Open.

The Open dialog box closes, and the path name including the file displays in the Open system file field.

4. In the FCU main screen, click the Insert  button.

The parameter line is registered into memory. The numbers on the right side of the status bar change from “2/2” to “3/3” to indicate three parameter entries are registered.

5. Click Save.

The parameter line is not saved until you click Save.




Verifying the Contents of a Parameter Line

You can verify the contents of a parameter line by clicking the Parameter

line  button.

Modifying the Contents of a Parameter Line

You can change the contents of a parameter line to correct mistakes and provide updated data.


1. Click the Parameter line  button to view the contents of the line.
2. Make changes as necessary.
3. Click the Replace  button to update the contents of the parameter definition file Parameter definition file being edited on the memory.
4. Click the Parameter line  button to verify that your changes have been registered.

Saving Parameter Definition Files to Disk

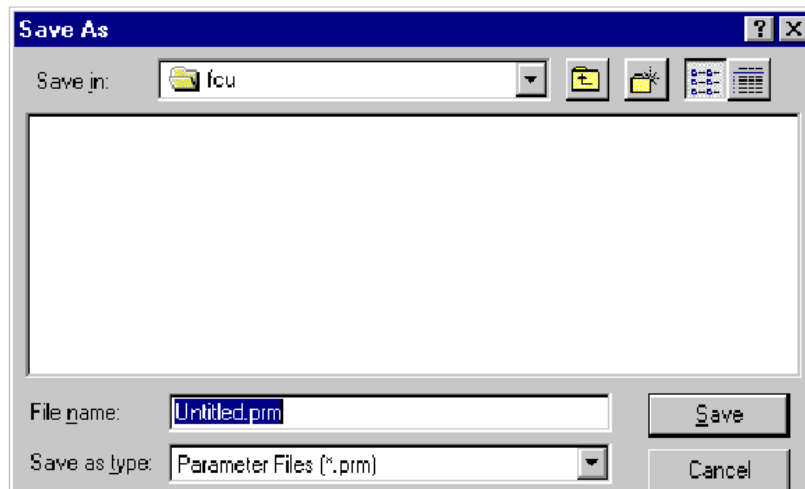
After you have successfully entered the parameter lines for the parameter definition file, you need to save the lines as a file on the Windows NT server.

Caution *If you do not save the lines as a file, the parameter changes are discarded as temporary information.*

To save parameter lines as a parameter definition file

1. In the FCU main screen, click the Save  button.

The Save as dialog box opens.




2. Type **Test.prm**, then click Save.

The Save as dialog box closes, and updates the name in the title bar to reflect the new file.

When you save the parameter definition file, the “end” line is not added as the last parameter line. This is still valid, because it can be omitted.

Opening Existing Parameter Definition Files

1. In the FCU main dialog, click the Open  button.

The Open dialog box opens.



2. Select the file you want to open, and click Open.

The parameter definition file displays in the FCU screen.

Executing Parameter Lines

You can execute parameter lines individually or consecutively. You can also view the error information and log files for the execution.

To execute parameter lines

1. In the FCU main screen, load the parameter line you want to execute as the first line of the parameter definition file.

2. Click the Option  button.

The Option dialog box opens.

3. At the bottom of the dialog box, select the Continuous execution check box to execute the lines in the parameter definition file simultaneously, or clear the check box to run a single parameter line.

4. Click Close.


The Option dialog box closes.

5. In the FCU main screen, click Execute.

The Execute dialog box opens for each line executed. The following example shows a normal termination.




To view the log file

- To view the log file for an execution, click the Log  button.

To view the log file for only the current execution, in the Option dialog box, before executing the file, select the Clear log file before execution check box. All other logs are cleared when you click Execute.

To view the error information

- To check the normal termination of each execution on the screen when executing multiple lines, click the Error Information  button before executing the lines.

The Error information dialog box opens. You can position the dialog box anywhere on the screen so that it does not overlap the FCU main screen during execution.

FCU OPERATIONS AND SCREENS FOR UNIX

This chapter describes the interface and operation instructions for using File Conversion Utility (FCU) with UNIX.

Copyright Screen

When you start the FCU, the Copyright and Operation screens open. After a few seconds, the copyright screen automatically closes.

The title and program versions vary depending on the environment.

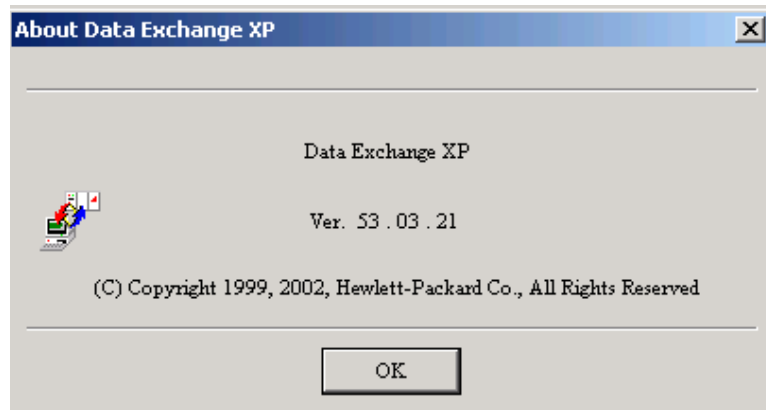


Figure 1. Copyright screen

Operation Screen

When you start the FCU, the Operation screen opens. You can set or change the parameters and start the file conversion operation in the Operation screen.

File Conversion Utility	
File	Help
Parameter File	<input type="text"/>
Volume File	<input type="text"/>
Direction	<input checked="" type="radio"/> M to O <input type="radio"/> O to M
Input File	<input type="text"/>
Output File	<input type="text"/>
	<input type="checkbox"/> PIPE <input type="button" value="OK"/> <input type="button" value="Cancel"/>
Code Conversion	<input type="radio"/> Ek<> J <input type="radio"/> Ec/E<>A <input checked="" type="radio"/> No <input type="radio"/> FILE <input type="text"/>
Padding	<input type="radio"/> Yes <input checked="" type="radio"/> No
Delimiter	<input type="radio"/> CR <input type="radio"/> LF <input checked="" type="radio"/> No
Emp	<input type="radio"/> Yes <input checked="" type="radio"/> No
RDW	<input type="radio"/> Yes <input checked="" type="radio"/> No
VSE/VOS1	<input type="text"/>
Status	<input type="text"/>

Figure 2. Operation screen

Operation Screen Menus

The Operation screen contains two menus:

File menu (page 142)

Help menu (page 143)

File Menu

Click **File** to access all processes other than inputting starter parameters.

Click any place other than on the menu to close it.

Load Submenu (Ctrl+L)

Click Load to read the starter parameters from the parameter definition file and display them respectively in each parameter entry field on this screen.

- When the parameter definition file is not created or no valid information is found in the parameter definition file, this specification is ignored.
- When the parameter file exists and any valid parameters are found, they display in the initial screen.
- Each time you click **Load**, the next parameter set loads, returning to the beginning after the last parameter set.

Save Submenu (Ctrl+S)

Click Save to save the starter parameter set currently input/displayed to the parameter definition file, replacing the parameter set read by clicking Load.

- When the parameter definition file does not exist, a new file is created to store the parameter.
- While the final parameter displays in the param window, if you enter a new parameter and click Save, the new parameter replaces the parameter previously displayed.

Delete Submenu (Ctrl+D)

Click Delete to delete the current loaded parameter set from the parameter definition file.

- If no parameter set is loaded, this selection is ignored.

Exit Submenu (Alt+F4)

Click Exit to terminate utility processes.

Help Menu

Click Help to display help information.

Click any place other than on the menu to close it.

Vol_tbl Submenu (Ctrl+T)

Click Vol_tbl to view the contents of the volume definition file.

‘*’ indicates the volume is read only.

Example 3390-3B can only be used as MTO, and 3390-3B is a read-only volume for open system.

Refer to “Volume Definition Files” (page 83) for outline of the volume definition file.

MF-File Submenu (Ctrl+M)

Click MF-File to display the list of dataset names of mainframe files.

To use this function, the volume serial number must be specified in advance in the following entry field:

“Input File” ----- Direction: MTO

“Output Field” ----- Direction: OTM

When you click MF-File, the information for each dataset in the volume specified by the input file displays.

Dataset Information: VSN = xxxxxx		Device Emulation Type = 3390-3B			
VSN Identification = MVS1					
Dataset Name	DO	RF	BL	RL	DS
*SAMFILE01.FIX	SAM	F	4096	128	150
-DAMFILE.F	DAM	F	4096	128	30
*SAMFILE02.VAL	SAM	V	4000	80	50
-PAMFILE	PAM	F	5000	100	200
-VIRTUALSTORAGEACCESS	VSAM	V	32768	4096	3000
-UNDEFSAMFILE	SAM	U	8000	200	80
-SAMFILESPANED	SAM	S	8192	8192	300

Figure 3. A list of the mainframe dataset

Figure Details

Dataset Name Indicates the name of the dataset.

A symbol “*” or “-” before a dataset name indicates the following:

*: Indicates that the dataset can be processed by the FAL/FCU.

-: Indicates that the dataset cannot be processed by the FAL/FCU.

?: Indicates that the dataset is unknown because necessary information for FCU is not defined. After the record format, record length, and block length are correctly defined in VSE parameter, the dataset can be processed.

DO Denotes the dataset organization type.

SAM: Indicates sequential access method organization

	DAM: Indicates direct access method organization
	PAM: Indicates partitioned access method organization
	VSAM: Indicates virtual storage access method organization
	???: Indicates unrecognized access method organization
RF	Denotes the record format.
	F: Indicates fixed-length record
	FB: Indicates fixed-length and blocking record
	V: Indicates variable-length record
	VB: Indicates variable-length and blocking record
	U: Indicates record of undefined length
	S: Indicates spanned record
	?: Indicates all other unidentified records
BL	Denotes block length (in bytes)
RL	Denotes record length (in bytes)
DS	Denotes dataset size (in tracks)

When the head volume of multiple volumes is specified, the total of all volumes is indicated.

When GUI environment is not available, the list of mainframe dataset names can be displayed by using LISTVOL function. Refer to “LISTVOL” (page 156) for the operation.

UX-File Submenu (Ctrl+U)

Click **UX-File** to display a list of file names for confirmation under the directory of a UNIX file to be output.

- When a directory name, followed by “/”, is specified as follows in the file name entry field, a list of file names displays in the specified directory.

“Output File” field --- MTO

“Input File” field --- OTM

- When a directory name is specified in the file name entry field, file names in the specified directory displays.
- When no directory name is specified in the file name entry field, file names in the current directory displays.

Example

UNIX FILE LIST: DIR = /aaaaa/bbbbbb/ccccc		
dddddd.dd	eeeeeee	ffffff.ffffff
hhhhh.hhhh	zzzzzz.z	xxxx.x
yyyyyyyyyy		

Error Submenu (Ctrl+E)

Click Error to display description of the most recent error.

OnVersion Submenu (Ctrl+O)

Click OnVersion to display the FCU version information.

Operation Screen Buttons, Check Boxes, and Fields

This section describes the Operation screen buttons, check boxes, and fields.

Parameter File Field

Input the parameter definition file name.

- When the default parameter definition file (fcudata.param) exists, the concerned file name displays.

Volume File Field

Displays the volume definition file name.

- The file name is **datasetmount.dat**.

Direction Buttons

Select the data transfer direction.

M to O From mainframe dataset to open system file.

O to M From open system file to mainframe dataset.

Input File Field

Type the name of the dataset or file in this field, using the following guidelines:

Caution *The parameter definition file must exist before you can enter the dataset file name in this field.*

If the data transfer direction is:

MTO	OTM
<p>Designate the mainframe dataset name using the following format: “volume/serial number:dataset name”, where:</p> <ul style="list-style-type: none"> • “volume-serial-number” is always six digits long • “dataset-name” is an alphanumeric character string up to 44 digits • A colon is placed between them <p>Example:</p> <p>DK7CE0:SAM.F80</p>	<p>Designate a name of the open system file to be input.</p> <ul style="list-style-type: none"> • If the file is created under the current directory, specify only the file name: samfile.fix80 • If the file is created under the specific directory, you can also specify the directory name and file name from the home directory by separating with a “/”: directory name under the home directory/rectory name/file name”.

Output File Field

The output dataset name or file name displays when the parameter definition file has already been created.

If the data transfer direction is MTO:

1. Designate a name of the mainframe dataset to be output
2. (*With named pipes only*) Designate a name of the named pipe to be output
3. If the designated file already exists, the new open system file overwrites the existing file from the beginning.

A confirmation message displays in the Status field.

If the designated file is not found, the utility generates a new file and writes the data in it.

4. Click OK to confirm the overwrite.
5. Click Cancel to stop the operation.

If the data transfer direction is OTM:

1. Designate a name of the open system file to be output.

The output file is always overwritten. No dataset is newly created or automatically expanded.

PIPE Check Box

Select the check box when the parameter definition file already exists and “PIPE=YES” is specified in the parameter definition file.

- Select the check box when using a named pipe.

In addition, when the data transfer direction is MTO, the check box indicates whether or not to use a named pipe by the data forwarding.

- Do not select the check box when performing data transfer to a file.

Code Conversion Buttons

Click one of the three Code Conversion buttons to indicate whether or not the data entities require code conversion.

Caution *Clicking the Ek<->J button converts code to Japanese.*

1. If code conversion is not required, or if you are using Data Exchange OTO, click No.

Data Exchange OTO does not support code conversion. An error is reported if you click a button other than No.

2. If code conversion is required, click Ec/E<->A to convert code from EBCDIC to ASCII format, or vice versa.

Depending on the status of code conversion, either Ec/E<->A, No, or FILE is already selected when the parameter definition file already exists.

3. If you are using the code conversion table file, input the file name in the FILE field.

Use the same format to specify the file name as the format used for UNIX file name for the input file or output file.

Padding Buttons

Click Yes or No to specify whether or not padding is required.

- If padding is required, click Yes.

If padding is not required, click No.

If the parameter definition file already exists, Yes or No is automatically selected based on whether the file indicates padding is required.

When the data transfer direction is:

MTO	OTM
When the mainframe dataset's record format is a variable-length, the FCU pads in shorter data entities so that all UNIX file data entities are the same length.	When the mainframe dataset's record format is a fixed length, the FCU pads in shorter data entities so that all dataset data entities are the same length.

Delimiter Buttons

Click CR, LF, or No to indicate whether or not delimiters are included.

When the parameter definition file already exists, CR, LF, or No is automatically selected based on whether the file indicates delimiters are

required. corresponding to the specification of delimiter in the Parameter definition file.

When the data transfer direction is:

MTO	OTM
<ol style="list-style-type: none">1. If a carriage return (CR) or line feed (LF) delimiter is required, click CR or LF to add the delimiter when data is written into UNIX file. When you click CR or LF, the delimiter is written into the UNIX file after each data entity.2. If delimiters are not needed, click No.	<ol style="list-style-type: none">1. If a carriage return (CR) or line feed (LF) delimiter is used, click CR or LF to remove the delimiter when it is read from the UNIX file. When you click CR or LF, data before the delimiter is extracted as a record and written into mainframe dataset, but is not transferred.2. Click No to transfer the delimiter as a part of the data entity. You cannot click No if padding is required. The delimiter must be included in the UNIX file. See “Padding Buttons” (page 150). Refer to “Code Converting Functions” (page 58) for the data editing format at the data transfer.

Emp Buttons

Click Yes or No to indicate whether or not to process the empty file during the file conversion process.

- Click Yes to process the empty file.

Click No if you do want to process the empty file.

RDW Buttons

Click Yes or No to indicate whether or not to add the RDW (record length of each record) for the file conversion process.

1. Click Yes to add the RDW.
2. Click No if you do not want to add the RDW.

VSE/VOS1 Field

Input the VSE dataset parameters (i.e., a dataset for which RF, RL, and BL are not defined). Separate the record format, record length, and block length by commas “,”.

Caution *If you are using Data Exchange OTO, do not enter parameter information in this field.*

If the parameter definition file already exists, the VSE parameter designated in the parameter definition file is automatically displayed.

Status Field

The Status field indicates the file conversion process' progress or error information.

Now checking	Displays during dataset search or attribute check execution.
Start	Displays at the start of the file conversion process. Unless the mainframe OS is VSE, progress is displayed in percentage (%) format during the file conversion process.
Complete	Displays at the completion of the file conversion process.

Error information displays when an error occurs and the process terminates abnormally.

OK and Cancel Buttons

To return to the Main FCU screen at any time before you start the conversion process, click Cancel.

Caution *After you start the file conversion process, you cannot cancel the request.*

1. When you are done entering data, click OK to start the file conversion process.

If the dataset contains data, a confirmation message displays, asking if you want to overwrite existing data.

Caution *If you execute the FCU without changing the parameters in the Operations screen, or if you click OK more than once, a warning displays, asking you to confirm the execution. Click Execute to start the process.*

2. Click OK to confirm the overwrite.

Caution *Do not click the OK button more than once! If you click the OK button again after starting the file conversion, another conversion process is queued and activated after the completion of the first conversion.*

Non-window FCU Operations

When the operation screens are not used, that is either **-nw** is specified in the initiation parameter or **fcunw** is invoked for UNIX platform, the operation proceeds as follows.

Indication and Confirmation of Detail Parameters

The FCU indicates the detail parameters obtained from the parameter on the window in which the initiation command was input. It also requests the acknowledgment response for it.

After checking the parameters, if you judge you may continue with the file conversion process, type OK to start the conversion process, or click Cancel display the detail of the next parameter and request the acknowledgment response.

When the end of detail parameters are detected, the process is terminated. When you specify **-nc** in the initiation parameter, no request for acknowledgment is made, but rather starts the file conversion process immediately after indicating parameters.

File Conversion Utility Ver. version name mnto VSN: dataset name UNIX file name CC PAD DEL ok/cancel ?

Figure 4. Confirmation of parameters

File Verification Display


FCU displays “Now checking” while verifying whether or not the file exists.

Confirmation of Overwrite

When the output UNIX file already exists, this function enables verifying whether or not to overwrite the file.

Type OK to overwrite the file, or type Cancel to stop the process.

Caution *When you specify **-nc** in the initiation parameter, the file conversion process (overwrite) starts without requesting a confirmation response for the overwrite.*



Overwrite ? ok / cancel ?

Figure 5. Confirmation of overwrite

File Conversion Process Initiation

FCU displays the message “Start” when the FCU process begins.

File Conversion Process Results

FCU displays the message “Complete” when the FCU process completes successfully.

When the process terminates abnormally, an error message displays.

End Statuses

- “0” is returned as an ending status when the process ends normally.
- “1” is returned as an ending status when the process ends abnormally.
- This ending status is included in “\$status” for C-shell, and “\$?” for B-shell/K-shell.

Completing the FCU Process

The following processes repeat until all the parameters in the parameter definition file are complete:

- “Indication and Confirmation of Detail Parameters” (page 154)
- “File Verification Display” (page 154)
- “Confirmation of Overwrite” (page 154)

The end status “0” is returned when the entire process completes successfully.

LISTVOL

This function displays a list of the dataset names of the specified volume.

Execution Method

LISTVOL is executed by using the following command and a parameter:

listvol VSN(VSN identification)

VSN This parameter specifies the volume serial number as a six alphanumeric characters

An error is returned if more than seven characters are specified

VSN identification Use a maximum of 35 alphanumeric characters

It is possible to omit this parameter.

Display Contents

Information within the volume specified with above command is displayed as shown below.

Caution *The volume definition file must be created in advance to use this function.*

# listvol xxxxxx,yyyy							
Dataset Name	DO	RF	RL	BL	TT	R	EX(Cyl:Trk)
*SAMFILE01.FIX	SAM	F	4096	4096	1	10	100:0
-DAMFILE.F	DAM	FB	128	4096	0	10	100:0
?SAMFILE.VSE	SAM	?	0	0	0	0	0:0

Figure 6. Contents of display by LISTVOL.

Dataset Name	<p>Displays the dataset name</p> <p>A symbol “*” or “-” placed before a dataset name indicates the following:</p> <p>*: Indicates that the FAL/FCU can process the dataset.</p> <p>-: Indicates that the FAL/FCU cannot process the dataset.</p> <p>?: Indicates that the dataset is unknown because necessary information for FCU is not defined.</p> <p>When the record format, record length, and block length are correctly defined in VSE parameter, the dataset can be processed.</p>
DO	<p>Displays the dataset organization type</p> <p>SAM: Indicates sequential access method organization</p> <p>DAM: Indicates direct access method organization</p> <p>PAM: Indicates partitioned access method organization</p> <p>VSAM: Indicates virtual storage access method organization</p> <p>???: Indicates other access method organizations not recognized by this utility</p>
RF	<p>Displays the record format</p> <p>F: Indicates fixed-length record</p> <p>FB: Indicates fixed-length and blocking record</p> <p>V: Indicates variable-length record</p> <p>VB: Indicates variable-length and blocking record</p>

U: Indicates undefined-length record

S: Indicates spanned record

?: Indicates an unidentified record other than above

RL Displays the record length (in bytes)

BL Displays the block length (in bytes)

TT+R: Last block address (The total of all volumes displays
when the head volume of multiple volume is specified.)

EX (Cyl:Trk) Data extent size (Number of cylinders:number of
tracks)

After the completion of LISTVOL command, the following returned value
will be reported:

0 Normal end

1 Error end

? Error Message:

When an error is detected during the operation, an error code and error
message are indicated in the standard error output and the error information
is written to the error log file.

Refer to “Definition Files” (page 83) for the detail.

FCU ERROR INFORMATION

The error codes and error messages output by the file conversion utility are shown below.

- The error codes having “*” may be reported when access contention between mainframe and open system occurs on the same volume. If the cause of the error cannot be identified from the contents in the following error message list, please check if any illegal contention on the shared volume has occurred.
- The error codes reported in LISTVOL command are also included within this table.
- An error code with plus value means a system error. The system error code is defined by the standard error file, “**errno.h**”, for UNIX system. In the case of Windows NT, it is defined in the “**errno.h**” and “**winerror.h**” being attached with **Microsoft Visual C++**.

UNIX FCU Errors

Table 1. UNIX FCU error codes and error messages

Error code	Error Message
-100	No parameter file Parameter definition file has not been created. => When the parameter definition file is specified: Check that the parameter definition file exists and that the name is correct. => When the parameter definition file is not specified with initiation parameter “-nw”: Default parameter definition file (fcudata.param) cannot be found in the current directory. Create the parameter definition file in the current directory.
-101*	Parameter file: Open error An error occurred at opening parameter definition file. => You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.
-102*	Parameter file: Read error An error occurred at reading parameter definition file. => You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.
-103*	Parameter file: No valid data No valid parameter is found in the parameter definition file. => Check if the parameters are set correctly to the parameter definition file.
-107	Parameter file: CODE_CONV error Code conversion specification in the parameter definition file is other than No. => Check if the code conversion is correctly specified as No.
-108	Parameter file: PADDING error Padding specification in the parameter definition file is other than Yes/No. => Check if the padding is correctly specified as Yes/No.

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-109	Parameter file: DELIMITER error Delimiter specification in the parameter definition file is other than CR/No. => Check if the delimiter is correctly specified as CR/LF/No or CRLF/No.
-110*	Parameter file: Open error An error occurred at opening and outputting the parameter definition file. => You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.
-111*	Parameter file: Write error An error occurred at writing the parameter definition file. => You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.
-112*	Parameter file: Close error An error occurred at closing the parameter definition file. => You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.
-114	Parameter: No input file name No parameter input file name is specified. => Specify parameter input file name.
-115	Parameter: VSN error Input VSN in the operation screen was incorrect. => Check the digit number in VSN.
-116	Parameter: Input file name error Input file name in the operation screen was incorrect. => Check if the input file name is correct.
-117	Parameter: Dataset name error Input dataset name in the operation screen was incorrect. => Check if the dataset name is correct.

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-118	<p>Parameter: Output file name error Parameter output file name is incorrect.</p> <p>=> Check if the parameter output file name is correct.</p>
-119*	<p>Input file: Open error An error occurred at acquiring dataset attribute information of input file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p> <p>If an error code of FAL is indicated, check the error content in table 2 (page 40). e.g., An actual partition name is different from that in the volume definition file, the error code = 6 (No such device) is indicated.</p>
-120	<p>Overwrite ? (OK/Cancel) This message is to confirm whether or not to overwrite the file.</p> <p>=> An open system file is specified as an existing output file. Select “OK”, to overwrite the file. Otherwise, select “Cancel” to specify another file.</p>
-121	<p>Output file: File name error Output file name is not specified.</p> <p>=>Check and specify the correct output file name.</p>
-122*	<p>Output file: Open error An OPEN error occurred when checking the output file is existing.</p> <p>=>You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-124	<p>Volume definition: MFtype error Incorrect MFtype is specified in the volume definition file.</p> <p>=> Specify MFN or MFA in MFtype of the volume definition file.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-125*	<p>Volume definition: VSN error Incorrect VSN is specified in the volume definition file.</p> <p>=> You can display the contents of the volume definition file by selecting “Volume” in HELP menu, and check a digit number of the VSN and check the VSN name in the specified parameter.</p>
-126	<p>Volume definition: Partition name error Incorrect partition name is specified in the volume definition file.</p> <p>=> You can display the contents of the volume definition file by selecting “Volume” in HELP menu, and check that the partition name begins with “/dev/rdisk”.</p>
-127*	<p>Volume definition: Emulation type error Incorrect device emulation type is specified in the volume definition file.</p> <p>=> You can display the contents of the volume definition file by selecting “Volume” in HELP menu, and check if the device emulation type is correct.</p>
-128*	<p>Volume definition file: Open error An error occurred at opening the volume definition file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p> <p>(e.g., If the volume definition file is not generated, an error code = 2 (No such file or directory)).</p>
-129*	<p>Volume definition file: Read error An error occurred at reading the volume definition file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-130*	<p>Volume definition file: No data No valid information is found in the volume definition file.</p> <p>=> You can display the contents of the volume definition file by selecting “Volume” in HELP menu.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-131*	<p>Volume definition file: Close error An error occurred at closing the volume definition file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-132	<p>Volume definition file: No valid error Incorrect parameter is specified in the volume definition file.</p> <p>=> Check all parameters in the volume definition file.</p>
-135	<p>Parameter error: No input file name Input file name (VSN) is not specified in the MF-File setting in Help.</p> <p>=> Specify VSN in the Input File.</p>
-136	<p>Parameter error: VSN error Input file name (VSN) is incorrect in the MF-File in Help.</p> <p>=> Check if VSN in the Input File has 6-digit number.</p>
-137	<p>Dataset error: No dataset No dataset is found while the MF-File is specified in Help.</p> <p>=> An volume which has no dataset is specified. Check if the VSN is correct.</p>
-138*	<p>Dataset error: Search error An error occurred in search of dataset.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>
-139	<p>Dataset error: Close error An error occurred at closing the dataset.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-140	<p>Input file error: Invalid organization type Dataset organization type of the mainframe is other than SAM.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the organization type of the dataset.</p>
-141	<p>Input file error: Invalid record format Record format of the mainframe dataset is not fixed or variable length.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the record format of the dataset.</p>
-142	<p>Input file error: Invalid block length Block length of the mainframe dataset is invalid.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the block length of the dataset. (Length of “0” or longer than 32 KB cannot be processed.)</p>
-143	<p>Input file error: Invalid record length Record length of the mainframe dataset is invalid.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the block length of the dataset. (Length of “0” or longer than 32 KB cannot be processed.)</p>
-144*	<p>Input file error: No data No dataset is found in the mainframe dataset.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the dataset size.</p>
-150*	<p>Input file: Open error An open error occurred in the input mainframe file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-151*	<p>Output file: Open error An open error occurred in the output UNIX file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-152	<p>Output file: Get file data error A data acquisition error of output file occurred in “OTM” operation.</p> <p>=> Collect information such as error log for troubleshooting.</p>
-153	<p>Processing data: Length check error A data length to be processed in “OTM” direction does not match.</p> <p>=> Collect information such as error log for troubleshooting.</p>
-155	<p>Buffer: Memory allocation error Memory allocation is failed.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
160*	<p>Input file: Read error A read error occurred in the input Mainframe file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>
-161*	<p>Output file: Write error A write error occurred in the output UNIX file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-162	<p>Output file: Code conversion error An error occurred in the code conversion to the output file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40)</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-163	<p>Get processing data error Getting process data failed.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40)</p>
-165	<p>Dataset error: Invalid data Invalid record length was found in a dataset.</p> <p>=> Check if the dataset is correctly generated from the mainframe side.</p>
-170*	<p>Input file: Close error File close error occurred in the input mainframe file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>
-171*	<p>Output file: Close error File close error occurred in the output UNIX file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-180	<p>UNIX/Open system file: Invalid directory name Incorrect directory name is specified in UX-File or NT-File setting in Help.</p> <p>=> Check the directory name and file name by using “list” command.</p>
-181	<p>UNIX file: Not a directory The specified name is not a directory name in the UX-File setting in Help.</p> <p>=> Check the directory name and file name by using “list” command.</p>
-182*	<p>UNIX/Open system file: Open Directory error A directory open error occurred in the UX-File or NT-File setting in Help.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-183	<p>UNIX/Open system file: Close directory error A directory close error occurred in the UX-File or NT-File setting in Help.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-190	<p>Input file name: No data No input file name is specified while the parameter save is instructed.</p> <p>=> Specify the “Input file name” before specifying “Save”.</p>
-191	<p>Output file name: No data No output file name is specified while the parameter save is instructed.</p> <p>=> Specify the “Output file name” before specifying “Save”.</p>
-192	<p>Parameter file name: No data Parameter file name is not specified.</p> <p>=> Specify the “Parameter file name”.</p>
-200	<p>Parameter file: End line The last parameter is loaded in the parameter file.</p> <p>=> By specifying “Load” again, the point will return to the top of the parameters.</p>
-201	<p>Parameter file: Direction error The data transfer direction specified in the parameter file is incorrect.</p> <p>=> Specify either MTO or OTM again.</p>
-202	<p>Parameter file: Too many data The number of parameter sets for parameter definition file exceeded 100.</p> <p>=> Parameter sets more than 100 cannot be processed. Decrease them to less than 100.</p>
-203	<p>Parameter: Empty select error The setting of “Emp=” is incorrect.</p> <p>=> Check if “Emp=” is set as “Emp=Yes” or “Emp=No”.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-204	<p>Parameter: RDW select error The setting of “RDW=” is incorrect.</p> <p>=> Check if “RDW=” is set as “RDW=Yes” or RDW=No”.</p>
-205	<p>RDW error: CODE_CONV unsupported Code conversion is not specified as “No” when “RDW=Yes”.</p> <p>=> The code conversion cannot be performed when “RDW=Yes”. Specify the code conversion to “No”.</p>
-206	<p>RDW error: PADDING unsupported Padding is not specified as “No” when “RDW=Yes”.</p> <p>=> Specify “No” for padding.</p>
-207	<p>RDW error: DELIMITER unsupported Delimiter is not specified as “No” when “RDW=Yes”.</p> <p>=> Specify “No” for delimiter.</p>
-210	<p>Parameter file: Comment line This is a comment line in the Parameter file.</p> <p>=> If specify the “Load”, it will move to the next line. It is also possible to replace with a valid parameter.</p>
-220	<p>Parameter: VSE select error The VSE parameter format is not correct.</p> <p>=> Check the number of VSE parameters and that “,” is correctly used as a separator among parameters.</p>
-221	<p>Parameter: VSE Record format error Record format in the VSE parameter is not correct.</p> <p>=> Check the Record format is set to one of F/FB/V/VB.</p>
-222	<p>Parameter: VSE Record length error Record length in the VSE parameter is not correct.</p> <p>=> Check if the Record length is set to a value within the extent allowed.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-223	<p>Parameter: VSE Block length error Block length in the VSE parameter is not correct.</p> <p>=> Check if the Record length is set to a value within the extent allowed.</p>
-230	<p>No Code conv. table file: No code conversion table. The code conversion table is not found.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-231	<p>Code conv. table: Open error The code conversion table could not open.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-233	<p>Code conv. table: Close error The code conversion table could not be closed.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-234	<p>Code conv. table: Get file data error The size of the code conversion table could not be obtained.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. Check the contents of the file.</p>
-235	<p>Code conv. table: File size error The size of the code conversion table is not correct.</p> <p>=> Check if the table size is 256 bytes.</p>
-236	<p>Code conv. table function: Invalid argument No source data to be converted was found.</p> <p>=> Check the contents of the input file, especially the delimiter.</p>
-238	<p>Code conv. table name: No data The file name of the code conversion table is not specified.</p> <p>=> Input the file name of the code conversion table, when “File” is selected as a code conversion method.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-250	<p>Named pipe: Create error The named pipe file could not be created.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-251	<p>Parameter : PIPE error The parameter “PIPE=” is incorrect.</p> <p>=> Check the parameter “PIPE=”. In the case of Data Exchange OTM, the pipe function is not supported. Set as “PIPE=No” or omit this parameter.</p>
-252	<p>Named pipe:Time out It doesn't become the status which can be written in a named pipe if it passes defined the time out value.</p> <p>=> Check if an application program or a utility program to receive data entries is running. Check if the time out value is correct.</p>
-253	<p>Named pipe: Wait time out value error The definition of the time out value is incorrect.</p> <p>=> Check if nWAIT_TIME_VALUE of the environment variable is less than 1441 and more than 0.</p>
-354	<p>Named pipe: Select error When process of PIPE ,error is reported in select of a file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-300	<p>Data error: Invalid record length The data length is not correct for Data Exchange OTM Padding function.</p> <p>=> Check if the data length of the input file is equal to or less than that of the dataset record length.</p>
-301	<p>Dataset error: Invalid record format The record format is not correct for Data Exchange OTM Padding function.</p> <p>=> Check if the record format of the dataset is fixed length.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-302	<p>Parameter error: Delimiter error The delimiter setting is not correct for Data Exchange OTM Padding function.</p> <p>=> Delimiter “No” cannot be specified for Data Exchange OTM Padding. Check if the Delimiter is correctly specified as “CR” or “LF”.</p>
-319*	<p>Dataset: Open error An error occurred at opening the dataset.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40). e.g., An actual partition name is different from that in the volume definition file, the error code = 6 (No such device) is indicated.</p>
-324	<p>O to M error: RDW unsupported “RDW=Yes” is specified when the transfer direction is OTM.</p> <p>=> Delete the RDW setting because the RDW option is not supported in OTM.</p>
-340	<p>Dataset error: Invalid organization type The organization type of the dataset is wrong.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the organization type of the dataset.</p>
-341	<p>Dataset error: Invalid record format Record format of the mainframe dataset is not fixed or variable length.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the record format of the dataset.</p>
-342	<p>Dataset error: Invalid block length Block length of the mainframe dataset is invalid.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the block length of the dataset. (Length of “0” or longer than 32 KB cannot be processed.)</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-343	<p>Input file error: Invalid record length Record length of the mainframe dataset is invalid.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the block length of the dataset. (Length of “0” or longer than 32 KB cannot be processed.)</p>
-350	<p>Input file: Open error An open error occurred in the input file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-351	<p>Output file: Open error An open error occurred in the output file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with Table 1.</p>
-352	<p>Input file: Get file data error A data acquisition error for input file occurred in OTM operation.</p> <p>=> Collect information such as error log for troubleshooting.</p>
-353	<p>Processing data: Length check error A data length to be processed in OTM direction does not match.</p> <p>=> Collect information such as error log for troubleshooting.</p>
-355	<p>Buffer: Memory allocation error Memory allocation is failed.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-360*	<p>Input file: Read error A read error occurred in the input file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-361*	<p>Output file: Write error A write error occurred in the output file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-362	<p>Output file : Code conversion error An error occurred in the code conversion to the output file.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the block length of the dataset.</p>
-363	<p>Get processing data error An acquisition of process data is failed.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-370	<p>Input file : Close error File close error occurred in the output file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-371*	<p>Output file : Close error File close error occurred.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals. If an error code of FAL is indicated, check the error content with table 2 (page 40).</p>
-379*	<p>UNIX file: No data No data is found in the input UNIX file.</p> <p>=> Specify a file that contains data.</p>
-380	<p>No UNIX/Open system file No UNIX/NT file specified is found.</p> <p>=> Specify the existing UNIX/NT file.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-381*	<p>UNIX/Open system file : Open error An open error occurred in the UNIX/NT file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-382	<p>Output file: Unsupported record format Record format of the output file is not supported.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the record format of the dataset.</p>
-383*	<p>Input file: Invalid format A format of the input file is wrong.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-384*	<p>Input file: Invalid delimiter position The delimiter position in the input file is not correct.</p> <p>=> You can display the attribute information of a dataset by selecting “MF-File” from the HELP menu, and check the record length of the input dataset. (Data record length of input file exceeds that of dataset, or a record with no data entity (i.e., data length = 4) is included.)</p>
-385	<p>Input file: File seeking error An error occurred in file seeking for input file.</p> <p>=> You can display a system error code by selecting “Error” from the HELP menu, and check the content of the error using OS manuals.</p>
-390	<p>Input file name: Length error An input file name in the parameter definition file is too long.</p> <p>=>Specify an input file name shorter than 1025 characters.</p>
-391	<p>Output file name: Length error An output file name in the parameter definition file is too long.</p> <p>=>Specify an output file name shorter than 1025 characters.</p>

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-392	Codeconv table: Length error A code conversion file name in the parameter definition file is too long. =>Specify a code conversion file name shorter than 1025 characters.
-393	VSE Parameter: Length error A VSE parameter in the parameter definition file is too long. =>Specify a VSE parameter shorter than 21 characters.
-394	Partition name: Length error A partitions name in the volume definition file is too long. =>Specify a partition name shorter than 1025 characters.
-395	VSN: Length error A VSN in the volume definition file is too long. =>Specify a VSN shorter than 7 characters.
-396	Emulation type: Length error An emulation type parameter in the volume definition file is too long. =>Specify an emulation type parameter shorter than 11 characters.
-397	Volume definition file: Length error The record length in the volume definition file is too long. =>Specify a record length shorter than 2081 characters (including delimiter).
-398	Parameter file: Length error The record length in the parameter definition file is too long. =>Specify a record length shorter than 3201 characters (including delimiter).
-399	Volume definition : VSN identification length error The VSN identification length in the volume definition file is too long. =>Specify a VSN identification length shorter than 35 characters.

Table 1. UNIX FCU error codes and error messages (continued)

Error code	Error Message
-400	Parameter : Invalid input file name An input file name is wrong. => Specify only one file name in the input parameter. (Multiple file names were specified.)

Windows NT FCU Errors

Table 2. Windows NT FCU error codes and error messages

Error code	Error Message
-100	Parameter definition file: Open error An error is detected at opening the parameter definition file. => Check if the parameter definition file is created. If the parameter definition file is created correctly, check the system error.
-101	Parameter: Count error An error is detected in the parameter count. => Check if the parameter count is correct.
-102	Parameter: Direction error Data transfer direction is not correct. => Check if the data transfer direction is specified correctly as MTO or OTM.
-103	Parameter: Mainframe file name error Mainframe file name is not correct. => Check if the mainframe file name is set correctly.
-104	Parameter: Open system file name error Open system file name is not correct. => Check if the open system file name is set correctly.
-105	Parameter: Code conversion error Code conversion setting is not correct => Check if the conversion is specified as “No”.
-106	Parameter: Padding error Padding setting is not correct. => Check if the Padding is specified as “Yes” or “No”.
-107	Parameter: Delimiter error Delimiter setting is not correct. => Check if the Delimiter is specified as “CRLF” or “No”.

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-108	Parameter: Add parameter error Delimiter setting is not correct. => Check if the adding parameter is correct.
-109	Parameter: Empty duplication error More than one empty setting is specified. => Specify only one "Empty setting".
-110	Parameter: RDW duplication error More than one RDW setting is specified. => Specify only one "RDW setting".
-111	Parameter: VSE duplication error There are more than one "VSE setting" is specified. => Specify only one "VSE setting".
-112	Parameter: VSE Record format error Record format of the VSE parameter is incorrect. => Set the record format to one of F/FB/V/VB.
-113	Parameter: VSE Record length error Record length in the VSE parameter is not correct. => Check if the record length is set to a value within the extent allowed.
-114	Parameter: VSE Block length error Block length in the VSE parameter is not correct. => Check if the block length is set to a value within the extent allowed.
-115	Parameter: VSE count error An error is detected in the VSE parameter count. => Check if the VSE parameter count is correct
-120	Volume definition file: Open error An error is detected at opening the volume definition file. => Check if the volume definition file exists. If the volume definition file exists, check the system error.

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-121	Volume definition file: Length error The record length in the volume definition file is too long. => Specify a record length shorter than 2080 characters (not including delimiter).
-122	Volume definition: Physical drive Length error A partitions name in the volume definition file is too long. => Specify a partition name shorter than 1025 characters.
-123	Volume definition: VOLSER Length error A VSN in the volume definition file is too long. => Specify a VSN shorter than 7 characters.
-124	Volume definition: Emulation type Length error An emulation type parameter in the volume definition file is too long. => Specify an emulation type parameter shorter than 11 characters.
-125	Volume definition: MFtype Length error Incorrect MFtype is specified in the volume definition file. => Specify MFN or MFA in MFtype of the volume definition file.
-126	Volume definition: VSN identification length error The VSN identification length in the volume definition file is too long. => Specify a VSN identification length shorter than 35 characters.
-130	Dataset: No dataset error No dataset is found. => Check if the mainframe file name is correctly specified, or if the dataset is correctly allocated on the specified volume.
-131	Dataset: Search error An error is detected in searching the dataset. => Check if the volume definition file name is specified correctly, and if the mainframe file name is specified correctly.

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-132	Dataset: Information get error An error is detected in acquiring dataset information. => Check if the volume definition file name is specified correctly, and if the mainframe file name is specified correctly.
-133	Dataset: Organization error A dataset organization specified is not correct. => Check if the dataset organization type is specified correctly.
-134	Dataset: Record format error A record format specified is not correct. => Check if the record format is specified correctly.
-135	Dataset: Block length error A block length specified is not correct. => Check if the block length is specified correctly.
-136	Dataset: Record length error A record length specified is not correct. => Check if the record length is specified correctly.
-137	Dataset: Dataset size error A dataset size specified is not correct. => Check if the dataset size is specified correctly.
-138	Dataset: Close error An error is detected during close operation. => Check the FAL error code and the system error code.
-150	Mainframe file: Open error An error is detected at opening the mainframe file. => Check the FAL error code and the system error code.
-151	Mainframe file: Read error An error is detected during reading data from the mainframe file. => Check the FAL error code and the system error code.

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-152	<p>Mainframe file: Write error</p> <p>An error is detected during writing data into the mainframe file.</p> <p>=> Check the FAL error code and the system error code.</p>
-153	<p>Mainframe file: Close error</p> <p>An error is detected at closing the mainframe file.</p> <p>=> Check the FAL error code and the system error code.</p>
-154	<p>Mainframe file: Record format error</p> <p>An error is detected in the Record format of the mainframe file.</p> <p>=> Data Exchange OTM Padding is specified against mainframe file that has a variable length of record format. Chose a fixed length mainframe file, or set the Padding off.</p>
-170	<p>Open system file: Open error</p> <p>An error is detected at opening the open system file.</p> <p>=> Check if the open system file name is correctly specified.</p> <p>Check the system error.</p>
-171	<p>Open system file: Read error</p> <p>An error is detected during reading data from the open system file.</p> <p>=> Check the system error.</p>
-172	<p>Open system file: Write error</p> <p>An error is detected during writing data into the open system file.</p> <p>=> Check the system error.</p>
-173	<p>Open system file: Close error</p> <p>An error is detected at closing the open system file.</p> <p>=> Check the system error.</p>
-174	<p>Open system file: No data error</p> <p>No dataset is found.</p> <p>=> Check if the open system file has data. Create appropriate data on the open system file.</p>

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-175	<p>Open system file: Delimiter (CR) position error Delimiter (CR) position error is detected.</p> <p>=> Check if the open system file name is correct. Check if the mainframe dataset name is correct. Check if the record length of the open system file is correct. (Data record length of input file exceeds that of dataset, or a record with no data entity (i.e., data length = 4) is included.)</p>
-176	<p>Open system file: Delimiter (LF) position error Delimiter (LF) position error is detected.</p> <p>=> Check if the open system file name is correct. Check if the mainframe dataset name is correct. Check if the record length of the open system file is correct.</p>
-177	<p>Open system file: Record format error An illegal record format is found.</p> <p>=> Check if the open system file name is correct. Check if the mainframe dataset name is correct. Check if the record format (fixed-length or variable-length) of the open system file data is correct.</p>
-178	<p>Open system file: Record length error An illegal record length is found. The data length of the open system file is over that of the mainframe file.</p> <p>=> Check the data length of the open system file and select a mainframe file which has the correct record length.</p>
-190	<p>Code conversion error An error was found during code conversion.</p> <p>=> Check if the dataset size is specified correctly.</p>
-200	<p>Process data get error An error is detected during close operation.</p> <p>=> Check the FAL error code and the system error code.</p>

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-220	<p>External table file: Open error The code conversion table could not be opened.</p> <p>=> Check the file name of code conversion table. Check the system error.</p>
-221	<p>External table file: Size error The code conversion table size is not correct.</p> <p>=> Check if the size is 256 bytes and if the table is correctly created.</p>
-222	<p>External table file: Read error A read error was found during reading the code conversion table.</p> <p>=> Check the system error.</p>
-223	<p>External table file: Close error The code conversion table could not be closed.</p> <p>=> Check the system error.</p>
-240	<p>Parameter: Direction, Padding, and Delimiter combination error The combination of the specified data transfer direction of OTM, padding of “Yes”, and delimiter of “No” settings is wrong.</p> <p>=> Specify the delimiter setting to “Yes” and padding to “Yes”.</p>
-241	<p>Parameter: Direction and RDW unmatched The combination of the specified data transfer direction of OTM and the setting of “RDW=Yes” is not allowed.</p> <p>=> Specify “RDW=No” if the transfer direction = OTM.</p>
-242	<p>Parameter: Code conversion and RDW unmatched The combination of “RDW=Yes” and code conversion setting other than “No” is not allowed.</p> <p>=> Specify “RDW=No” if the code conversion setting is other than “No”. If “RDW=Yes”, specify the code conversion setting to “No”.</p>
-243	<p>Parameter: Padding and RDW unmatched The combination of “RDW=Yes” and padding setting “Yes” is not allowed.</p> <p>=> Specify “RDW=No” if the padding setting is “Yes”. In case of “RDW=Yes”, specify the padding setting to “No”.</p>

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-244	<p>Parameter: Delimiter and RDW unmatched The “RDW=Yes” and Delimiter setting of “Yes” is not allowed.</p> <p>=> Specify “RDW=No” if the delimiter setting is “Yes”. If “RDW=Yes” specify the delimiter setting to “No”.</p>
-245	<p>Parameter: Specified VOLSER is not defined Volume Definition file. Specified VOLSER is not defined the volume definition file.</p> <p>=> Check whether specified VOLSER is defined in the volume definition file.</p>
-300	<p>Parameter definition file : Length error The record length in the parameter definition file is too long.</p> <p>=> Specify a record length shorter than 3200 characters (not including delimiter).</p>
-301	<p>Mainframe file name: Length error An input dataset name (for Data Exchange MTO) or an output dataset name (for Data Exchange OTM) in the parameter definition file is too long.</p> <p>=> Specify an input/output dataset name shorter than 1025 characters.</p>
-302	<p>Open system file name: Length error An input file name (for Data Exchange OTM) or an output file name (for Data Exchange MTO) in the parameter definition file is too long.</p> <p>=> Specify an input/output file name shorter than 1025 characters.</p>
-303	<p>Code conversion: Length error A code conversion file name in the parameter definition file is too long.</p> <p>=> Specify a code conversion file name shorter than 1025 characters.</p>
-304	<p>VSE : Length error A VSE parameter in the parameter definition file is not corrected.</p> <p>=> Specify a VSE parameter shorter than 21 characters.</p>
-305	<p>VSE record-format: Length error The record format for VSE in the parameter definition file is not corrected.</p> <p>=> Specify a record format for VSE shorter than 3 characters.</p>

Table 2. Windows NT FCU error codes and error messages (continued)

Error code	Error Message
-306	VSE record-length: Length error The record length for VSE in the parameter definition file is not corrected. => Specify a record length for VSE shorter than 6 characters.
-307	VSE block-length: Length error The block length for VSE in the parameter definition file is not corrected. => Specify a block length for VSE shorter than 6 characters.

OPERATIONAL NOTES

This chapter provides additional information about:

- “FAL/FCU Operations” (page 188)
- “IO Contention between Mainframe and Open System” (page 193)

FAL/FCU Operations

- FCU does not have functions to verify that the SAM file has completed on the mainframe side. The system must check for the file manually or provide some coordination functions between the mainframe and open system host systems. In addition, because the utility does not provide mutual exclusive control, the contention between mainframe and open system must be managed by an administration level control. Use the same caution for Data Exchange OTO operations.
- *(For Solaris only)* To correct the error “glibXm.so.xx is not found”, define a path to the Xmlibrary using the following operations.

When using C shell

For all others

Add the following line in the home directory **.cshrc**.

Add the following two lines in the home directory **.dtprofile**.

```
setenv LD_LIBRARY_PATH
/usr/dt/lib:$LD_LIBRARY_PATH
H
```

```
LD_LIBRARY_PATH=/usr/dt/lib
:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

When not in the common desktop environment, add the two lines to the file **.profile** in the home directory. If **.profile** does not exist, create it.

When you are done, log out and log back in.

- When the UNIX files used by FCU are located in devices other than XP256/512, complete the “mounting” before executing FCU.

If you execute FCU operation before completing the mounting, errors detected may not be reported to the FCU.

- The file conversion utility does not access mainframe side data area other than VTOC. Therefore, you cannot apply an access control by catalogue or security control functions.
- FCU can access datasets with standard VTOC. FCU ignores indexes for datasets containing an Index VTOC, processing the dataset the same way as for standard VTOCs. FCU accesses the dataset sequentially.

- Dataset names do not support spaces. When FCU detects a dataset name that contains spaces, it only accepts characters before the space as the dataset name, and continue the process.
- The partition size defined in the open system must be the same as the volume size in the mainframe. If the partition size is shorter than that of the mainframe, open system will not be able to access the extent of the volume.
- When Data Exchange volume (e.g. 3390-3B) is being installed by user ID = root (i.e. super user), generic users cannot use the FCU.

(UNIX only) If generic users need to use the FCU during this time, configure a group between an owner of Data Exchange volume (i.e. root) and the concerned user, and give the user access rights (see note below) for the Data Exchange volume.

(Windows NT only) Use administrator access rights to access the Data Exchange volumes.

- *(UNIX only)* Ignore any warning messages (shown below) if you start the FCU and request to use the operational screen. This does not affect the FCU function.



Warning

Missing characters in String to FontSet conversion



Warning

Cannot convert string “-dt-interface system-medium-r-normal
-m*_*_*_*_*_*_*_*” to type FontSet

- Except for Windows NT, the FAL/FCU process may terminate unsuccessfully when any of the following signals is issued. This can also force other FAL/FCU processes to remain open and make the used shared memory unreserved, occupying the memory space.

SIGUSR1	SIGSTOP
SIGUSR2	SIGTSTP
SIGILL	SIGCONT
SIGTRAP	SIGTTIN
SIGIOT	SIGTTOU
SIGABRT	SIGVTALRM
SIGEMT	SIGPROF
SIGFPE	SIGXCPU
SIGKILL	SIGXFSZ
SIGBUS	SIGWAITING
SIGSEGV	SIGLWP
SIGSYS	SIGFREEZE
SIGALRM	SIGTHAW
SIGPOLL	SIGCANCEL
SIGIO	

When this happens, use the “kill” command to cancel the processes and use “ipcrm” command to delete the shared memory areas that have KEY=0. Refer to manuals for each operating system.

- Versions 47.XX.00 and before do not support file sizes larger than 2GB. Executing a file larger than 2GB will generate an error.

- When configuring MSCS, note the following items.

(Windows NT only) Signatures must be written on the volumes under MSCS before configuring MSCS (The MSCS server cannot connect volumes without signatures)

(Windows NT only) Volumes that have signatures cannot be accessed from another server

(Windows NT only) Volumes that have signatures cannot be shared

(Windows NT only) Volumes that have signatures can only be accessed by the mainframe and the server that wrote the signature

(Windows 2000 only) Service Pack 1 must be installed when MSCS is configured

(Windows 2000 only) Signatures are not required for MSCS Configuration

- Data Exchange OTO does not support code conversion. An error is reported if you click a button other than No.
- When you find a problem in the data transferred by a named pipe, check the data using a named pipe that is not edited by an application or a utility program.

Example

dd if=pipe-file of=outfile

pipe-file: The pipe name to be output to the pipe

outfile: The file name to store data on the pipe

1. FAL/FCU uses shared memory (**shmget()**) less than 1048577 byte and more than 8 bytes. Specify **YYYYY<=8 and XXXXX>=1048576**, if the following item is added in etc/system file. (the following item is not indispensable)

set shm:shminfo_shmmax=XXXXX

set shm:shminfo_shmmin=YYYYY

2. Do not access the HMED volume from FAL/FCU, when other system (UNIX/NT/OS2/MF) is accessing it. Do not access the Data Exchange volume from another system when FAL/FCU is accessing it. The above

limitation was canceled in the one after the version 48.XX.20 (Except for AIX and Windows NT). The above limitation was canceled in the one after the version 50.XX.YY for Windows NT.

- Data Exchange VOLUME does not support alternate passes.
- 3. Do not update the volume that is transferred by Data Exchange OTM directory.
- 4. Wait a little to access the Data Exchange volume, after Data Exchange OTM from Windows NT. (The time of delay changes with a server's performance.)
- 5. Reboot the Windows NT machine after the open volume using Data Exchange is expanded.
- 6. Do not access the Data Exchange volume using command of server OS (ex:dd command). It is only FAL that it can access the Data Exchange volume.
- 7. A patch is necessary when the open system is AIX and Fibre channel and User ID is not root. As for the details, ask IBM.
- 8. When the mainframe OS is VSE, DFSORT cannot be used after the Data Exchange volume is copied to another volume. Copy using Ditto when copying Data Exchange volume to the other volume.
- 9. Do not write a signature on Data Exchange volumes that emulation type is 3390-3X or 3380-KX (X=A,B,C) under Windows 2000 environment. If you tries to write a signature on it by Disk Administrator with Windows 2000, a Write Error appears to avoid to write the signature. When the Disk Administrator of Windows 2000 starts again, it asks you to write the signature again, then please ignore that message not to write the signature.

(Solaris only) When using OTO data transfer and OPEN-L/M and LUSE volume, the number of cylinders used is smaller than another OS.

(Solaris only) The data cylinder must be less than or equal to 32767, so the geometry parameter is differ from another volumes. For information, refer to the Solaris Configuration Guide.

IO Contention between Mainframe and Open System

The Data Exchange volumes belong to mainframe volumes. Mainframe can access it as same as other mainframe volumes. Open system is allowed to access the dataset in the volume only through FAL/FCU programs. Under the Data Exchange volume being reserved by mainframe, an access from open system terminates abnormally.

An exclusive control to access of Data Exchange volumes need to be implemented at users administrative operation to avoid the access contention on the same volume. And some job coordination between mainframe and open system is also required in order to exchange data between them.

1. An access to a dataset created on the Data Exchange logical unit must be done through FAL/FCU.
2. A concurrent access to the Data Exchange volume from mainframe and open system becomes the following result:

Under the reserved condition by mainframe

- An FAL/FCU access terminates unsuccessfully.
- Accesses from open system except for read or write IO are executed successfully.
- Accesses from mainframe are executed as same way as mainframe environment.

Note:

- An access from the open system may terminate successfully if open system can retry successfully after the reserve is released by the mainframe. Because the time interval which an open system can wait with retry varies depending upon OS, and it also depend upon mainframe application which issues reserve, FAL/FCU cannot guarantee the successful access.

Under the un-reserved condition by mainframe

- Accesses from both mainframe and open system can be executed.

3. "Reserve" issued by open system does not affect mainframe IO. For example, AIX server uses the SCSI "Reserve" command to exclude IOs issued by other systems. Data Exchange volumes which are reserved by open system can execute mainframe IO. "Reserve" is effective only among the open systems

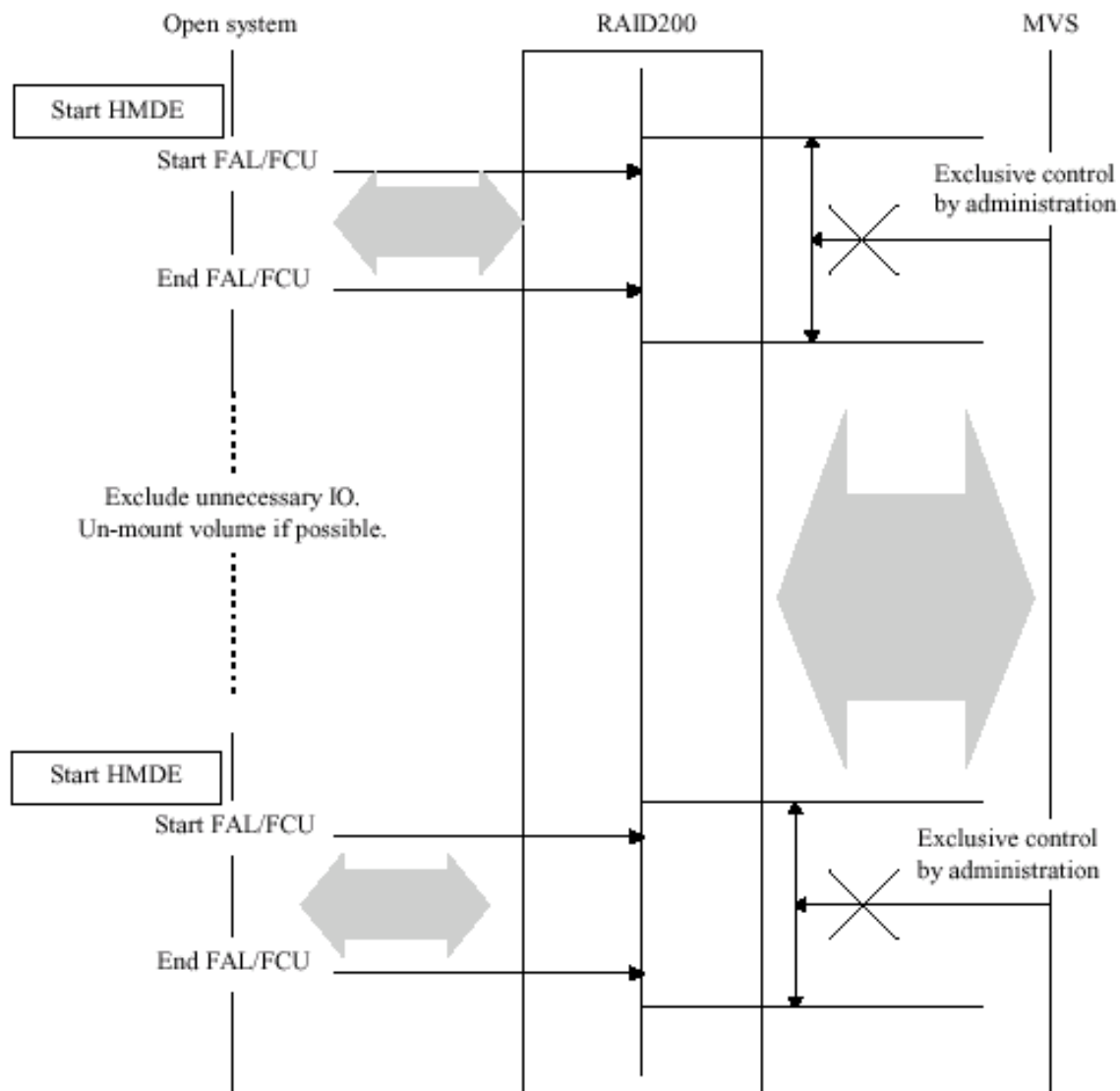


Figure 1. Data Exchange exclusive control

TROUBLESHOOTING

When a problem occurs and you cannot identify the cause, collect the relevant information for your platform and contact your technical support team.

Troubleshooting for UNIX

- Error codes (FCU, FAL, and SYS): Displayed in the command line and by clicking Error in the Help menu.
- Parameters in FCU: Displayed in the command line and in the Operation window.
- Contents of current volume definition file.
- Contents of parameter definition file, if it is used.
- Command line log, if possible.
- When an error is detected in the FAL, an error log is generated. When **ERR_LOG_FILE** is specified in the environment variables, the error logs is output into the file specified.

Example

“/tmp/fal_error” and “/tmp/fal_error.bak”

“/tmp/fal_dump” and “/tmp/fal_dump.bak”

- Contents of syslog, especially of error information.

Troubleshooting for Windows NT

- FCU log file: Recorded with the parameter definition file name and a .log extension.

Example Parameter definition file : **Test.prm**
 Log file: **Test.prm.log**

1. FAL log file:

Example **C:\fal_error** and **C:\fal_error.bak**
 C:\fal_dump and **C:\fal_dump.bak**

2. Dr. Watson's log file (when Dr. Watson's error messages appear at an error)

Example **DRWTSN32.LOG**

The log file is saved in the directory specified at Windows NT installation. For example, in C:\WINNT.

For more information, refer to Dr. Watson's online help for Windows NT.

DATA EXCHANGE OTO OPERATION

When using the Data Exchange OTO (open system to open system data transfer) function, use the following instructions to perform required operations.

Providing the Data Exchange OTO Operational Environment

Before starting Data Exchange OTO operation, prepare the following operational environment:

1. Define RAW devices to be used as the intermediate and shared volumes between open systems
2. Install formatter, allocator, and FAL/FCU programs
3. Create the volume definition file
4. Create intermediate volumes using formatter
5. Create files on the intermediate volumes using allocator
6. Create the parameter definition files (this is necessary to use FCU)
7. Compile programs that use FAL programs
8. Prepare files and data for data exchange and executing FAL/FCU

Defining RAW Devices

Those volumes to be used as intermediate volumes and to be shared between open systems must be defined as OPEN-3/8/9/K emulation type for XP256/512, and must be defined as “raw” devices from each host server. For open systems, there are no means to distinguish OPEN-3/ 8/9/K for open system dedicated volumes from these Data Exchange volumes. Make sure not to confuse the usage on those volumes in the host systems.

Prohibited Operations

Caution *Do not use following operations to create file systems on the intermediate volumes, or information on the volume may be destroyed and the volumes become unusable as Data Exchange volumes.*

- SUN Solaris
“newfs” command
- HP-UX
“pvcreate” command
- IBM AIX
creating a volume group
- Windows NT
formatting and creating a file system
- Digital UNIX/Tru64 UNIX
“newfs” command
- Sequent Dynix/ptx
creating a file system
- NCR SVR4
creating a volume group

Installing Formatter (FMT) and Allocator (ALC)

Use the same installation procedure as described in “Providing Operational Environment” (page 70).

Creating Intermediate Volumes

This operation creates the initial format of the intermediate volume to be used for FAL/FCU on the device defined as a raw device. This operation is executed by using formatter (FMT).

- The formatter and allocator are provided separately for UNIX and Windows NT.
- The size of the volume can be expressed with number of cylinders. The cylinder size is defined as below, but data capacity actually stored in it becomes smaller than the values below.

1 Cylinder = 15 Tracks

1 Track = 96 Sub-blocks

1 Sub-block = 512 Bytes

- The maximum number of cylinders allowed by formatter is the following

Emulation type	OPEN-3	OPEN-8	OPEN-9	OPEN-K	OPEN-E	OPEN-M
47.XX.00 or before	3331	5818	5818	2536	—	—
YY.XX.ZZ not LUSE	3331	5818	5818	2536	5818	5818
YY.XX.ZZ LUSE	5818	5818	5818	5818	5818	5818
YY.XX.20 not LUSE	3331	9959	10009	2536	19752	63992
YY.XX.20 LUSE number of volumes	Min(3338*n-7 ,65534)	Min(9966*n-7 ,65534)	Min(10016*n- 7,65534)	Min(2543*n-7 ,65534)	Min(19759*n -7,65534)	Min(“63999*n- 7,65534)

XX=01 or 02; YY: Number; ZZ<20

— Not supported Min (A,B): The one whose A and B are small

- The CVS (customizable volume size) feature can be applied to Data Exchange volumes including Data Exchange OTO. For OPEN-3/OPEN-8/OPEN-9/OPEN-K, the cylinder extent allowed for CVS is shown below.

(The value that the size specified with CVS is converted into the cylinder)–7

Example

Set 4096 MB for CVS when OPEN-8 or OPEN-9 and the volume can be used effectively

- When a wrong volume is formatted, the volume data is destroyed. Take a special caution when specifying parameters for formatter.
- Do not use formatter and allocator simultaneously. Start and use one of them after the other one is completed. When Windows NT is used, it must be started after the other one is completed and the operation window is closed.
- The volume definition file must be created before executing formatter and allocator.
- Be careful when ‘\’ is used for the volume serial number or the dataset name in UNIX system.

For example ‘\’ must be specified “\\” in Bourne shell, because ‘\’ is used for the escape character.

- After the volume is formatted by UNIX formatter 41.01.02, Windows NT allocator (42.01.02 or above) cannot create intermediate files. Format the volume using formatter 42.01.02 or above again

Formatter \ Allocator		YY.XX.ZZ (XX=01 or 02, YY>41, ZZ>19)		YY.XX.ZZ (XX=01 or 02, YY>41, ZZ<20)	
		UNIX	Windows NT	UNIX	Windows NT
#41	UNIX	OK	NG	OK	NG
	Windows NT	OK	OK	OK	OK
YY.XX.ZZ (XX=01 or 02, YY>41, ZZ<20)	UNIX	OK	OK	OK(*2)	OK(*2)
	Windows NT	OK	OK	OK(*2)	OK(*2)
YY.XX.ZZ (XX=01 or 02, YY>41, ZZ>19)	UNIX	OK	OK	NG(*1)	NG(*1)
	Windows NT	OK	OK	NG(*1)	NG(*1)

OK: It is possible to allocate

NG: Cannot allocate

(*1): It is not possible to Data Exchange OTM, Data Exchange MTO and listvol too

(*2): The number of the cylinders which can allocate is less than 4370

For UNIX OS

Use the following commands and parameters to execute formatter:

mfformat -d devname -v VOLSER-name [-p primary cylinders]

-d devname

Specify the raw device name on which the shared volume format is to be created. This setting cannot be omitted. Note that this is also true for volume definition file.

-v VOLSER-name

Specify the volume serial number by which the shared volume can be distinguished, in six alphanumeric characters (A-Z, @, #, \, and 0-9). Only capital letters are allowed; spaces and symbols are not allowed. This is also true for volume definition file.

-p primary cylinders

- Specify the number of primary cylinders within the extent of 2 through 5818 and in decimal value. This field can be omitted, except for CVS. The default value is maximum value is the following table.

Emulation type	OPEN-3	OPEN-8	OPEN-9	OPEN-K	OPEN-E	OPEN-M
47.XX.00 or before	3331	5818	5818	2536	—	—
YY.XX.ZZ (YY>47, ZZ<20)	3331	5818	5818	2536	5818	5818
YY.XX.ZZ (YY>47, ZZ<19)	3331	9959	10009	2536	19752	63992

XX=01 or 02

- Only one value is allowed in each field after **-d**, **-v**, and **-p**.
- When the formatter terminates unsuccessfully, an error message displays.

When the “Format check error” message displays, the formatting operation does not start, and the setting parameters remain the same.

For all other messages, the formatting operation starts but finds an error. Use Table 2 (page 220) and Table 3 (page 223) to determine which actions to take.

The following value is returned when formatter terminates.

0: successful termination

1: failed termination.

For Windows NT

Click the Format icon to open the Format dialog box.

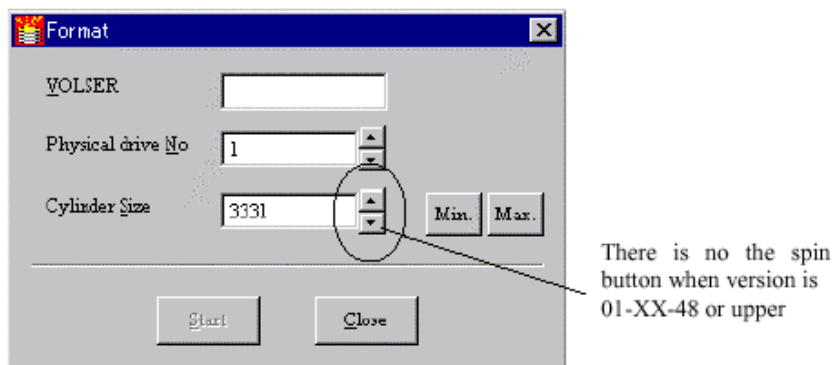


Figure 1. Format dialog box

VOLSER

Specify the volume serial number by which the shared volume can be distinguished, in 6-digit alpha numerical characters (A-Z, @, #, \, and 0-9). Only capital letters are allowed; spaces and symbols are not allowed. This number is also used to find the volume in allocator and to specify the VSN in volume definition file for executing FAL/FCU.

Physical drive No. Specify the device number in one or two digits numerical characters on which the shared volume format is to be created. This setting number is also used for physical drive number in the volume definition file which is used for FAL/FCU execution.

Cylinder No. Specify the number of primary cylinders within the extent of 2 through 5818 and in decimal value.

Min Specify the minimum value (i.e. 2) instead of setting the volume size in Cylinder No.

Max Specify the maximum value (i.e. 5818) instead of setting the volume size in Cylinder No.

Version 46.XX.YY or 47.XX.YY 5818

Version 48.XX.ZZ or upper

Emulation type	OPEN-3	OPEN-8	OPEN-9	OPEN-K	OPEN-E	OPEN-M
YY.XX.ZZ (YY>47, ZZ<19)	3331	9959	10009	2536	19752	63992
YY.XX.ZZ (YY>47, ZZ<20)	3331	5818	5818	2536	5818	5818

Start Initiate the formatter after setting all the necessary information. When you click this button, the Format dialog box below opens, and confirms you want to continue.

If you click OK, the volume data are initialized and original data is overwritten.

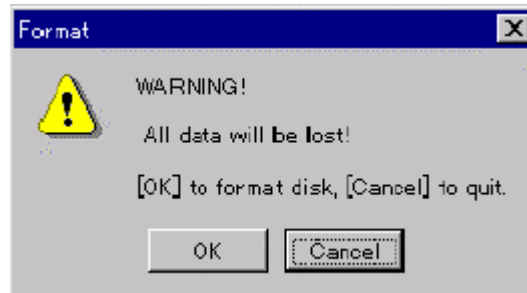


Figure 2. Format warning

When the specified VSN has already been used for another volume, the dialog box below opens.



Figure 3. VOLSER duplication error dialog box

When the formatting process has completed successfully, the dialog box below opens.



Figure 4. Normal end dialog

When the formatting process terminates with an error, the dialog box below opens.



Figure 5. Error end dialog box

Close Close formatter.

Creating Intermediate Files

You can create intermediate files on a volume that has already been initialized by formatter using allocator.

- You cannot use allocator on a volume that is not formatted.
- The data capacity that can be stored in the intermediate file is smaller than its physical capacity and varies depending upon the block length to be used. Check the available data capacity in one track with the table 1 (page 208) and calculate the necessary file size on the intermediate file. Note that when the variable length of record format is used, the intermediate file must have some margin in size, because an actual block size may become smaller than the defined one.

Table 1. The relation between block length and write available capacity per track

Block length by allocator (A) in bytes	Write available data per track (bytes)	Block length by allocator (A) in bytes	Write available data per track (bytes)	Block length by allocator (A) in bytes	Write available data per track (bytes)
23477 - 32760	$(A) \times 1$	1589 - 1684	$(A) \times 22$	565 - 596	$(A) \times 44$
15477 - 23476	$(A) \times 2$	1493 - 1588	$(A) \times 23$	533 - 564	$(A) \times 45$
11477 - 15476	$(A) \times 3$	1397 - 1492	$(A) \times 24$	501 - 532	$(A) \times 46$
9077 - 11476	$(A) \times 4$	1333 - 1396	$(A) \times 25$	469 - 500	$(A) \times 48$
7477 - 9076	$(A) \times 5$	1269 - 1332	$(A) \times 26$	437 - 468	$(A) \times 49$
6357 - 7476	$(A) \times 6$	1205 - 1268	$(A) \times 27$	405 - 436	$(A) \times 51$
5493 - 6356	$(A) \times 7$	1141 - 1204	$(A) \times 28$	373 - 404	$(A) \times 53$

Table 1. The relation between block length and write available capacity per track (continued)

Block length by allocator (A) in bytes	Write available data per track (bytes)	Block length by allocator (A) in bytes	Write available data per track (bytes)	Block length by allocator (A) in bytes	Write available data per track (bytes)
4821 - 5492	$(A) \times 8$	1077 - 1140	$(A) \times 29$	341 - 372	$(A) \times 55$
4277 - 4820	$(A) \times 9$	1045 - 1076	$(A) \times 30$	309 - 340	$(A) \times 57$
3861 - 4276	$(A) \times 10$	981 - 1044	$(A) \times 31$	277 - 308	$(A) \times 59$
3477 - 3860	$(A) \times 11$	949 - 980	$(A) \times 32$	245 - 276	$(A) \times 62$
3189 - 3476	$(A) \times 12$	917 - 948	$(A) \times 33$	213 - 244	$(A) \times 65$
2933 - 3188	$(A) \times 13$	853 - 916	$(A) \times 34$	181 - 212	$(A) \times 68$
2677 - 2932	$(A) \times 14$	821 - 852	$(A) \times 35$	149 - 180	$(A) \times 71$
2485 - 2676	$(A) \times 15$	789 - 820	$(A) \times 36$	117 - 148	$(A) \times 74$
2325 - 2484	$(A) \times 16$	757 - 788	$(A) \times 37$	85 - 116	$(A) \times 78$
2165 - 2324	$(A) \times 17$	725 - 756	$(A) \times 38$	53 - 84	$(A) \times 83$
2005 - 2164	$(A) \times 18$	693 - 724	$(A) \times 39$	21 - 52	$(A) \times 88$
1877 - 2004	$(A) \times 19$	661 - 692	$(A) \times 40$	1 - 20	$(A) \times 93$
1781 - 1876	$(A) \times 20$	629 - 660	$(A) \times 41$		
1685 - 1780	$(A) \times 21$	597 - 628	$(A) \times 42$		

“Write available data per track” includes 4 bytes information per record indicating record length, and 4 bytes information per block indicating block length, when a variable length is specified in the record format. Thus, if the record format to be used is a variable length, the area to store the information must be considered in addition to the user data.

“Write available data per track” $\times 15$ = data capacity per cylinder.

For UNIX OS

Use the following commands and parameters to execute allocator.

allocds {-d devname | -v VOLSER-name | -d devname -v VOLSER-name}[-n datasetname]

[-f F/FB/V/VB] [-r reclen] [-b blocklen] [-c cylinders]

<i>Command Details</i>	-d devname	Specify the raw device name that has been formatted by formatter. This setting cannot be omitted.
	-n VOLSER-name	Specify the volume serial number by which the shared volume can be distinguished, in six alphanumeric characters (A-Z, @, #, \, and 0-9). Only capital letters are allowed; spaces and symbols are not allowed. This is also true for volume definition file.
	-n datasetname	Specify the dataset name to be created in less than or equal to 44 alphanumeric characters (A-Z, @, #, \, 0-9) and period (.). When allocating an intermediate file, this setting cannot be omitted. When this setting is omitted, the residual capacity of the intermediate volume is indicated in number of cylinders.
	-f F/FB/V/VB	Specify the record format. This setting can be omitted. The default value is "F". F: Fixed-length and deblocking record FB: Fixed-length and blocking record V: Variable-length and deblocking record VB: Variable-l
	-r reclen	Specify the record length in decimal within an extent of 1 to 32760. This setting cannot be omitted. The default value is 4096.

Record format = F: Record length = Block length

Record format = FB: Record length = Block length / N
(N: Integer)

Record format = V/VB: Record length = Block length -
4 or smaller and 5 or longer.

-b blocklen Specify the block length in decimal within an extent of
1 to 32760. This setting can be omitted.

(default) Record format = F/FB: Block length = Record
length

(default) Record format = V/VB: Block length =
Record length + 4

Record format = F: Block length = Record length

Record format = FB: Block length = Record length × N
(N: Integer)

Record format = V/VB: Block length = Record length +
4 or longer

-c cylinders Specify the dataset size in number of cylinders in
decimal. This setting can be omitted and the default
value is 100 cylinders.

When the format version is YY.XX.ZZ (XX=01 or 02,
ZZ<20), the number of the cylinders that is 4369 or
available capacity is small can be specified.

When the format version is YY.XX.ZZ (XX=01 or
02, ZZ>19), available capacity

Only one value can be specified after each field of -d, -n, -f, -r, -b, -c.

When the allocator terminates unsuccessfully, an error message is
indicated.

If “Allocate check error” message displays, the allocator process does not start and the original conditions remain.

For all other messages, an error has been detected during allocator process. Use the following guidelines to remove an error and run allocator again.

- When the allocator terminates successfully, the complete message and number of unused cylinders is indicated.
- Devname and VOLSER-name specify both or either one side. Specify both Devname and VOLSER-name to prevent a specification mistake.
- The following values are returned when formatter terminates:
 - 0: successful termination
 - 1: failed termination

For Windows NT

Select an Allocator icon and initiate the Allocator. When the intermediate volume created by formatter already exists in the system, the Operation screen in figure 6 (page 213) displays.

Figure 6. Allocator operation screen

- VOLSER** Specify the volume to be processed by the allocator among intermediate volumes being formatted by formatters. The VSN can be directly specified in this field or can be selected in the pull-down list on the screen.
- Dataset** Specify the dataset name to be created in less than or equal to 44-digit alpha numerical characters (A-Z, @, #, \, 0-9) and period (.).
- Allocation Capacity** Specify the file size of the intermediate file to be created.
- The total of the number of the cylinders and the tracks that can be specified is the following.
- When format version is YY.XX.ZZ (XX=01 or 02, ZZ<20): the number of the cylinders that is 4369, or available capacity is small can be specified.

	When format version is YY.XX,ZZ (XX=01 or 02, ZZ>19): Available Capacity
Cylinder	Specify the file size to be created in number of cylinders. The number of tracks can be specified together.
Track	Specify the file size to be created in number of tracks. The file size equals the total number of cylinders and tracks.
Max.	Specify the maximum size instead of using cylinder and/or track. Entire area indicated in "Available Capacity" field is allocated for the file space.
Record format	Specify the record format. One of the following types can be directly specified or can be selected from the pull-down list on the screen. F: Fixed-length and deblocking record FB: Fixed-length and blocking record V: Variable-length and deblocking record VB: Variable-length and blocking record
Record length	Specify the record length in decimal. The following extents are allowed. F: Record length = Block length FB: Record length = (Block length) / n , n: integer V: 5 = Record length = (Block length - 4) VB: 5 = Record length = (Block length - 4)
Block length	Specify the block length in decimal within an extent of 1 to 32760. Record format = F/FB: 1 = Block length = 32760

Record format = V/VB: 9 =Block length =32760

When you click Copy, the record length values are set as the block length values.

Start Start the allocator after setting the necessary information.

When the allocator terminates successfully, the dialog box below opens.

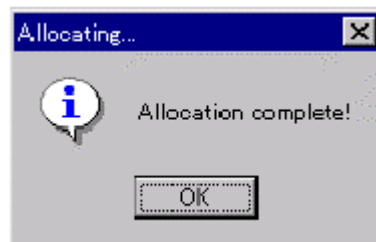


Figure 7. Normal end dialog box

When the allocator terminates unsuccessfully, the dialog box below opens.



Figure 8. Error end dialog box

Errors

When “Allocate check error” displays in the Allocator dialog box, the allocating operation does not start and the original condition remains. For all other messages, use the following guidelines to remove the error condition and allocate the file again.

- Open error! (n): Open process error on the intermediate volume.
- Seek error! (n): Seek process error on the intermediate volume.
- Read error! (n): Read process error on the intermediate volume.
- Write error! (n): Write process error on the intermediate volume.
- Close error! (n): Close process error on the intermediate volume.

Caution *Do not click DS List. It does not work.*

For additional error message details, refer to “Formatter/ Allocator Error Information” (page 219).

No Existing Data Exchange Volumes

If the intermediate volume does not exist, the screen indicating no Data Exchange volume in figure 9 (page 216) appears. When the screen indicating no Data Exchange volume appears, check that the formatter has correctly executed on the volume.



Figure 9. A screen when no Data Exchange volumes found

Creating Volume Definition Files

The same operation as defined in Data Exchange MTO and Data Exchange OTM is applied. Refer to “Volume Definition Files” (page 83).

Creating Parameter Definition File

The same operation as defined in Data Exchange MTO and Data Exchange OTM, except for the notes below, is applied. Refer to “Parameter Definition Files” (page 95).

Specify no for code conversion functions, or the result of conversion cannot be guaranteed. The definitions related to VSE in the section are not applied to Data Exchange OTO and do not appear on the screen.

Starting Data Exchange Operation

The data exchange operation to be executed for FAL and FCU for Data Exchange OTO is also the same as explained in each relating sections for Data Exchange MTO and Data Exchange OTM. Refer to “Using the File Access Library Application Program” (page 51) for FAL operations and “Providing Operational Environment” (page 70) for FCU operations.

Formatter/ Allocator Error Information

The error messages output from the formatter and the allocator are shown below.

Errors in UNIX

Table 2. Formatter and Allocator error message in UNIX

Error message		Meaning
F O R M A T T E R	Format Failed...	The formatter terminates unsuccessfully.
	Format check error	An error was found with the format check. Check the number of primary cylinder specified.
	VOLSER-name error	Incorrect characters are specified in the Volume name
	Primary cylinders error Primary_cylinders =[2-5818]	The primary cylinders is not the range that it can be specified, or it is invalid letter
	The number of cylinder is too large	There is no capacity specified cylinders
	Available volume is not found for formatting	The Volume that device emulation type is OPEN-3/8/9/K does not exist.

Table 2. Formatter and Allocator error message in UNIX (continued)

Error message		Meaning
A L L O C A T E R	Allocating dataset failed...	The allocator terminates unsuccessfully.
	Allocating dataset failed... Allocate check error	An error was found with the allocator check. Check the number of cylinder specified.
	Allocate check error	An error was found with the allocator check. Check the number of cylinder specified.
	Record_format error	The specified record format is not correct.
	Record_length error	The specified record length is not correct.
	Record_length=[The range that it can be specified]	
	Block_length error	The specified block length is not correct.
	Block_length=[The range that it can be specified]	
	Cylinders error	The number of cylinders is specified characters except for the numerical value or '0'.
	Dataset name error (too long)	A specified dataset name exceeds 44 characters.
	Dataset name error (Invalid letter)	Invalid characters are specified or the dataset name isn't specified.
	This device is not formatted	A specified device is not formatted.
	Dataset is full	It exceeds the number of the data set which can be created.
	This dataset is already exists	The same dataset name has already existed.
	The number of cylinder is too large	There is no capacity specified cylinders.
	Available volume is not found for allocating	The Volume that device emulation type is OPEN-3/8/9/K does not exist.
	The VSN of allocating volume is disagreement	VSN on the volume is inconsistent with specified VSN.(or on volume definition file)

Table 2. Formatter and Allocator error message in UNIX (continued)

Error message		Meaning
C O M M O N	Volume definition file is not found	Volume definition file is not found
	Volume definition file has no data	Volume definition file has no data
	The devname and/or VOLSER-name is not found in volume definition file	Specified the device name. or the volume name or those combinations does not exist in the file.
	Partition name length error in volume definition file	A partitions name in the volume definition file is too long.(less than 1025 characters)
	VSN length error in volume definition file	A VSN in the volume definition file is too long. (less than 7 characters)
	Emulation type length error in volume definition file	An emulation type parameter in the volume definition file is too long.(less than 11 characters)
	Volume definition file record length error	The record length in the parameter definition file is too long.(less than 2081 characters)
	Volume definition file read error	A read error occurred in the volume definition file (Start again)
	Volume definition file close error	File close error occurred in the volume definition file (Start again).

Errors in Windows NT

Table 3. Formatter and Allocator error message in Windows NT

	Error message	Meaning
F O R M A T T E R	Format Failed...	The formatter terminates unsuccessfully.
	Format check error	An error was found with the format check. Check the number of primary cylinder specified.
	Available volume is not found for formatting	The Volume that device emulation type is OPEN-3/8/9/K does not exist.
A L L O C A T E R	Allocating dataset failed...	The allocator terminates unsuccessfully.
	Allocating dataset failed... Allocate check error	An error was found with the allocator check. Check the number of cylinder specified.
	There is no volume for allocating!	There is no formatted volume.
	This VOLSER is already exists	The same VOLSER name has already existed
	This dataset is already exists	The same dataset name has already existed.
	Available volume is not found for allocating	The Volume that device emulation type is OPEN-3/8/9/K does not exist.
	Format information is error	Information on the specified volume is incorrect. Format the volume.
	The VSN of allocating volume is disagreement	VSN on the volume is inconsistent with specified VSN.(or on volume definition file)

Table 3. Formatter and Allocator error message in Windows NT (continued)

Error message		Meaning
C O M M O N	Volume definition file is not found	Volume definition file is not found
	Volume definition file has no data	Volume definition file has no data
	The devname and/or VOLSER-name is not found in volume definition file	Specified the device name. or the volume name or those combinations does not exist in the file.
	Partition name length error in volume definition file	A partitions name in the volume definition file is too long.(less than 1025 characters)
	VSN length error in volume definition file	A VSN in the volume definition file is too long. (less than 7 characters)
	Emulation type length error in volume definition file	An emulation type parameter in the volume definition file is too long.(less than 11 characters)
	Volume definition file record length error	The record length in the parameter definition file is too long.(less than 2081 characters)
	Volume definition file read error	A read error occurred in the volume definition file (Start again)

INDEX

A

- allocator
 - errors in UNIX 220
 - errors in Windows NT 223
 - installing 201

C

- code conversion table 58
 - user-defined 58
- code converting functions 58
- code systems supported 22

D

- data capacity 16
- Data Exchange
 - multiple volume datasets 67
- data transfer direction 21
 - bi-directional 68
 - OTO 200
- datasets
 - acquiring attribute information 33
 - closing 32
 - multiple volume 67
 - opening 26
 - reading data 28
 - writing data 30
- delimiter adding functions 62

E

- errors
 - FAL 32
 - FCU 159
 - formatter/allocator 220, 223
 - information 39

F

FAL

- acquiring dataset attribute information 33
- acquiring error information 32
- closing datasets 32
- compiling 48
- creating intermediate volumes 202
- creating volume definition files 48
- deinstalling for UNIX 77
- deinstalling for Windows NT 80
- error information 39
- features 7
- functions 25
- inputting license keys 81
- installing 48
- installing for UNIX 72
- installing for Windows NT 79
- opening datasets 26
- operational environment 48
- operations notes 188
- program version updates 8
- reading data 28
- setting license keys 82
- shared volume definition 48
- supported objects 16
- system configurations 14
- using 51
- writing data 30

FCU

- creating intermediate volumes 202
- creating new parameter definition files in Windows NT 128
- creating volume definition files in Windows NT 125
- deinstalling for UNIX 77
- deinstalling for Windows NT 80
- errors 159
- features 7
- inputting license keys 81
- installing for UNIX 72
- installing for Windows NT 79
- interface for Windows NT 110
- main screen buttons and fields for Windows NT 123
- main window toolbar 121
- opening parameter definition files 137
- operational environment 70
- operations for Windows NT 125
- operations notes 188
- pipe functions 22
- program version updates 8
- saving parameter definition files 136
- setting license keys 82
- starting 105
- starting for Windows NT 107, 126
- supported objects 16
- system configurations 14
- toolbar for Windows NT 121
- UNIX non-window operations 154
- using code conversation table 58
- Windows NT main screen 110

File Access Library. *See* FAL.

file conversion functions 54

File Conversion Utility. *See* FCU.

formatter
 errors in UNIX 220
 errors in Windows NT 223
 installing 201
functions
 code converting 58
 data editing formats 54
 delimiter adding 62
 FAL 25
 file conversion 54
 padding 60
 pipe 56
 record length adding 66

I
IO contention 193

L
license keys
 inputting for FAL/FCU 81
 setting for FAL/FCU 82

M
maximum data capacity 16
multiple volume datasets 67
multiple volume definition files 93

O
open system to open system. *See* OTO. 21
Operation screen
 buttons, check boxes, and fields 147
 menus 142
 UNIX 141
operational environment
 FAL 48
 FCU 70
 OTO 200

OTO
 creating parameter definition files 217
 creating volume definition files 216
 operational environment 200
 operations 199
 starting operations 218
 using the manual 21

P
padding functions 60
parameter definition files 95
 creating in FCU 128
 opening existing 137
 OTO 217
 saving to disk in FCU 136
parameter lines
 executing 137
 modifying content 135
 verifying content 135
pipe functions 22, 56
 time out value 56

R
record length adding functions 66

S
server machines supported 22
shared volumes 48, 70
supported objects 16
 code systems 22
 data transfer directions 21
 host (mainframe) datasets 17
 host (mainframe) OS 17
 mainframe device emulation types 19
 pipe functions 22
 server machines 22
system configurations 14

T

time out value

- pipe functions 56

toolbar

- FCU main window for Windows NT 121

troubleshooting

- UNIX 196

- Windows NT 197

U

UNIX

- copyright screen 140

- deinstalling FAL/FCU 77

- FAL/FCU features 7

- FCU errors 160

- formatter/allocator errors 220

- installing FAL/FCU 72

- non-window FCU operations 154

- Operation screen 141

- setting license keys 82

- starting FCU 105

- troubleshooting 196

V

version updates 8

volume definition file

- creating 48

volume definition files 83, 216

- creating for FCU 125

- multiple 93

W

Windows NT

- creating new parameter definition files in
FCU 128

- creating volume definition files 125

- deinstalling FAL/FCU 80

- errors 178

- FCU interface 110

- FCU main screen 110

- FCU main screen buttons and fields 123

- FCU operations 125

- formatter/allocator errors 223

- installing FAL/FCU 79

- opening FCU parameter definition files 137

- saving FCU parameter definition files to
disk 136

- setting license keys 82

- starting FCU 126

- troubleshooting 197